A perspective to incident response

or another set of recommendations for malware authors

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy - *TLP:WHITE*

alexandre.dulaunoy@circl.lu

June 7, 2013

CIRCL, national CERT of Luxembourg

- CIRCL[1] is composed of 6 full-time incident handlers + 2 FTE backup operators.
- The team is operating as an autonomous technical team relying on its own infrastructure.
  - Operators competencies include reverse engineering, malware analysis, network and system forensic, software engineering and data mining.
- CIRCL, the national CERT, is part of SMILE[2] gie (a publicly funded organization to promote information security in Luxembourg).
- In 2012, CIRCL handled more than 10000 security events and conducted more than 400 technical investigations.

[1] `http://www.circl.lu/`

[2] `http://www.smile.public.lu/`

Disclaimer

Even if the presentation includes recommendations for malware authors, the main objective is to share techniques used by the attackers and especially how to detect these techniques within a targeted infrastructure.

Like any secure coding recommendations, I don't expect these to be read a lot... by the malware authors.

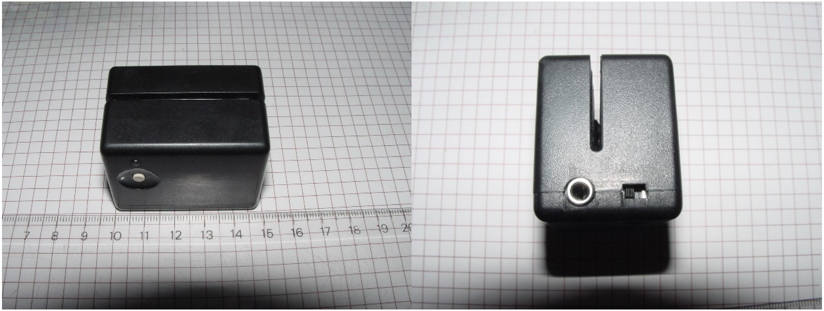
The origin of the malware author recommendations...

- As a (potential) malware author, you might wonder why a CERT is giving recommendations. It could be a way to force malware authors to go in specific directions...
- An incident response team is usually encountering a lot of malware. Humans tend to compare what they see and analyse.
- Malware is often just a piece of software. Sometime is a clever piece of software and sometime it's just crap[3]. Some are using clever tricks or some not.
- The recommendations are just a collection of what we saw as analysts and where improvements[4] could be.

[3] Don't worry security software can also fall into this category.

[4] or a trend for a security researcher

Learning from old school criminals



- Keep it simple...

Code signing

Should I be scared, as a malware author, about code signing?
What are my competitors doing? How do they sign their malware?

Stealing private keys

- Various compromised systems have accessible certificate store. A standard "PFXExportCertStoreEx with EXPORT_PRIVATE_KEYS flag"[5] can do the job to gather private keys.
- If you cannot steal private keys, you can still purchase stolen private keys from some colleagues running SpyEye/Zeus/Citadel campaigns.

[5] http://www.circl.lu/pub/tr-13/

Asking the CA to sign

- Another option is to ask a CA (or sub-CA) to sign your code.
- You have around 600 CAs/sub-CAs around the world. You might find the one that is economicaly or politically close to you.
- As some CAs are just checking the domain name, a stolen subdomain could do the job.
- Revocation of your certificate might be an issue but it's not uncommon to see weeks or months before the revocation is effective in CRL or in OCSP[6].

[6] Assuming X.509 revocation process is properly working at your target

Are CA Trustworthy on a revocation side?

What are the revocation reasons of X.509 certificate? After one year of fetching X.509 CRL, you can have a good overview:

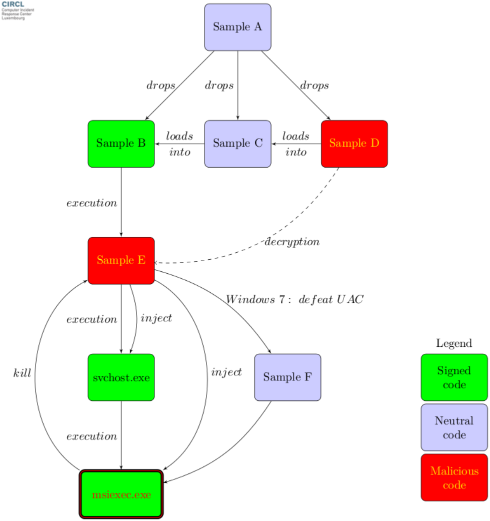
Hits	Revocation Reason
678039	Cessation Of Operation (code 5)
172888	Unspecified (code 0)
89823	Certificate Hold (code 6)
88788	Superseded (code 4)
76445	Key Compromise (code 1)
43482	Affiliation Changed (code 3)
3910	Privilege Withdrawn (code 9)
230	CA Compromise (code 2)
1	A A Compromise (code 10)

Compromising the CA

- In the set of 600 CAs/sub-CAs, you can find one (or more) CA vulnerable. Especially some sub-CAs have weaknesses in their web interfaces.
- Some attackers compromised various CAs[7] at different levels.
- A lot of work? As an attacker, you are looking for an easier and faster path.

[7] Not only DigiNotar if you look at the reason of revocation in the CRLs, CA compromise is not uncommon.

Using legitimate signed binaries - PlugX case



Using legitimate signed binaries - PlugX case

- The easiest path is to take existing signed binaries and find "vulnerabilities" where you can load your binaries from the running signed binaries.
- In the case of PlugX[8], they used various legitimate DLL (from McAfee to Symantec products) to abuse the LoadLibrary function.
- As the signature verification is only done when loading, the signed binary can be then used in memory to load the malicious payload.
- Revocation is unlikely to happen as the DLL is used in legitimate software. If the vendor needs (wants) to fix its software, it will take some time.

[8] http://www.circl.lu/pub/tr-12/

Network communication

The don't(s) when you write your communication protocols for your malware.

Binary protocols, PoisonIvy...

- PoisonIvy (RAT) protocol is relying on a binary protocol.
- Many default parameters like TCP Port 3460, fixed protocol size and default password.
- For an incident response team, it's simple to scan, detect or even brute-force for PoisonIvy
  - PoisonIvy victim sends 256 bytes → Controller
  - Controller response with 256[9] bytes → PoisonIvy victim
  - Controller send 4 bytes (size to be send) but hardcoded 0xd0150000 → PoisonIvy victim

[9] 0x35e1066ccd15873eeef8518966b70f8b first 16 bytes - with default admin password

Binary protocols, Linux sshd library rootkit...

- A recent Linux malware rootkits (aka cPanel/libkeyutils.so.1.9) the ssh library to gather username/password.
- The author(s) used UDP packet on port 53 to send exfiltred username/password.
- It seems very clever (to hide your traffic in DNS traffic) but...
- NIDS are trying to decode the DNS payload without success. (→ error on decoding)

HTTP protocol, why not? but?

- A good start to be embedded in the traffic but you should avoid protocol done like Fakem RAT:
- Where the RAT client is sending an HTML page to the server via HTTP and with a constant title.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(content:"<html><title>1"; depth:14; content:"6<|2F|title><body>";)
```

- Avoid to use meaningful name like:

  ```
  http://<IP>:1001/c.php?botnet=<victimname>
  ```

MiniDuke, some good practises

- MiniDuke[10] is a set of targeted attacks composed of a set of malware.
- Initial bootstrap of the malware used social networks to fetch next stage of the malware.
- Proxy C&C used known compromised system. The compromised systems were multihomed virtual hosts (6000 hosts).
- From a network analysis perspective, victims are checking IP addresses → generating a lot of false-positives.

[10] http://www.circl.lu/pub/tr-14/

HTTP good pratices for malware authors

- Using a random list of User-Agent headers for a malware is not very clever. (e.g. some companies analyse the distribution of UA agents per workstation)
- Instrumenting the web browser is usually more efficient and limit detection.
- Don't forget that latency and time schedule are critical when "instrumenting" web monkeys.
- As an example, Snifula is using standard DeleteUrlCacheEntry()[11] to delete the URLs from the browser history.

[11] http://www.circl.lu/pub/tr-13/

Domain and hostname management for your C&C

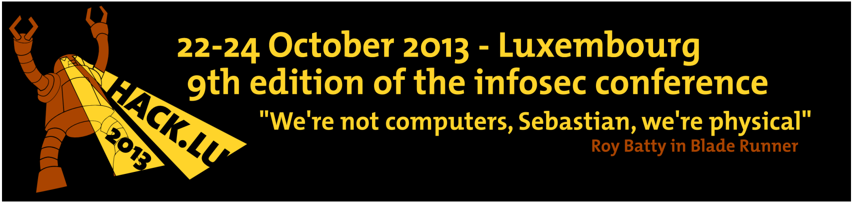
- Don't re(use) the same domains for various targets.
- Don't share the same IP addresses on various hostnames. (Passive DNS are great and especially for incident response team)
- Don't use the same email addresses for a set of domains to be used by your C&C.
- Don't forget that DNS is full of record types. (A record is just one type, you might want to use TSIG, SOA, SPF,...)

Conclusion

- The list of recommendations is not exhaustive. (e.g. memory usage versus disk usage, custom crypto, logging, debug, VM detection,...)
- Malware authors should not underestimate OPSEC[12].
- Don't forget that the mess is on both side and can be from the benefit of the other side.
- Sometime the attacker can be the victim, don't forget about it when you write your software.

[12] urlhttp://www.slideshare.net/grugq/opsec-for-hackers

Q&A?


HACK.LU
2013
22-24 October 2013 - Luxembourg
9th edition of the infosec conference
"We're not computers, Sebastian, we're physical"
Roy Batty in Blade Runner