Anatomy of Exploit Kits

Preliminary Analysis of Exploit Kits as Software Artefacts

Vadim KOTOV   Fabio MASSACCI

{kotov, massacci}@disi.unitn.it
University of Trento, Italy

ESSoS 2013

Table of Contents

1. Drive-by-download attacks problem
2. What is drive-by-download attack?
3. Our contribution
4. Structure of an exploit kit
5. Offensive component
6. Defensive component
7. Management component
8. Source code
9. Conclusions and future work

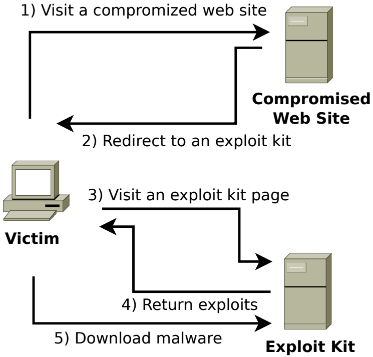
Drive-by-download attacks problem (2012)

Kaspersky Lab's statistics*

Rank	Name	No. attacks	% of all attacks
1	Malicious URL	1 393 829 795	87.36%
2	Trojan.Script.Iframer	58 279 262	3.65%
3	Trojan.Script.Generic	38 948 140	2.44%
4	Trojan.Win32.Generic	5 670 627	0.36%
...	...	...	...

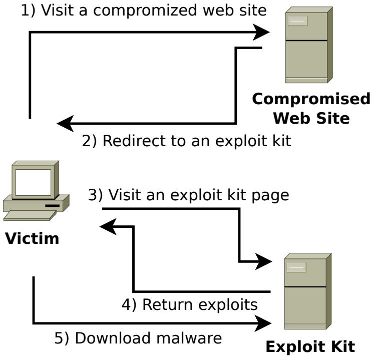
*http://www.securelist.com/en/analysis/204792255/Kaspersky_Security_Bulletin_2012_The_overall_statistics_for_2012

What is drive-by-download attack?


1) Visit a compromized web site
Compromised Web Site
2) Redirect to an exploit kit
Victim
3) Visit an exploit kit page
4) Return exploits
5) Download malware
Exploit Kit

1. Victim loads a compromised web site with an *iframe* pointing at the malicious URL
2. The attacking page is about to be loaded within the iframe
3. Victim sends an HTTP request to the malicious server (without knowing it)
4. An HTML document containing exploits is loaded within the *iframe*
5. An exploit downloads and starts a malware executable

What is drive-by-download attack?


1) Visit a comprimized web site
Compromised Web Site
2) Redirect to an exploit kit
Victim
3) Visit an exploit kit page
4) Return exploits
5) Download malware
Exploit Kit

1. Victim loads a compromised web site with an *iframe* pointing at the malicious URL
2. The attacking page is about to be loaded within the iframe
3. Victim sends an HTTP request to the malicious server (without knowing it)
4. An HTML document containing exploits is loaded within the *iframe*
5. An exploit downloads and starts a malware executable

A server application that stands behind the drive-by-download attack is called EXPLOIT KIT

Our contribution

Analysis of exploit kits as *software artefacts*

Our contribution

Analysis of exploit kits as *software artefacts*

The data

- Leaked source codes of 30+ exploit kits
- Vulnerability and exploit information on 70+ kits

Our contribution

Analysis of exploit kits as *software artefacts*

The data

- Leaked source codes of 30+ exploit kits
- Vulnerability and exploit information on 70+ kits

Main results

- The attacks are not that sophisticated as expected
- Exploits are outdated and affected software is limited
- Profit is to come by large numbers

Structure of an exploit kit

Offensive component

- Fingerprint victim machines
- Exploit vulnerabilities

Defensive component

- Evade AV scanners detection
- Hide from search robots

Management component

- Report statistics
- Provide configuring options

Code protection

- Prevent unauthorized distribution
- Complicate analysis

Offensive component

Interesting observations

- The workflow of the attack is more or less the same in all the kits
- 88% of the exploit kits perform simple browser and operating system detection
- 64% also block repeating IP addresses
- 36% of kits throw attacks even if victim's browser and OS are unsupported

Offensive component

Interesting observations

- The workflow of the attack is more or less the same in all the kits
- 88% of the exploit kits perform simple browser and operating system detection
- 64% also block repeating IP addresses
- 36% of kits throw attacks even if victim's browser and OS are unsupported

Vulnerability analysis

- In average an exploit kit has around 10 exploits
- Most of the exploits in a kit are 1-2 years old
  - Malware authors prefer using public exploits rather than 0-day?
  - Marketing a new exploit is time-consuming?
- Most exploited apps are: Flash, Java, MSIE, Reader

Defensive component

What possibly can it be

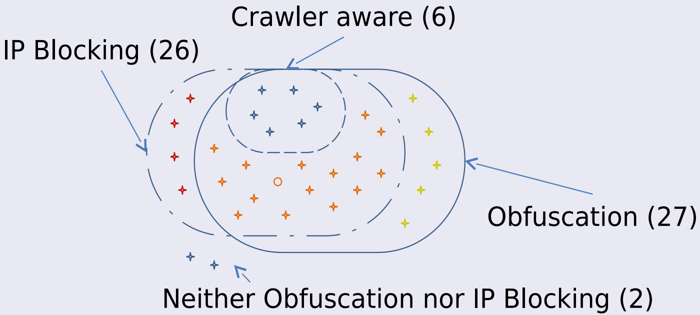
- Trick antivirus signatures using obfuscation (82%)
- Block repeating IP to prevent probing by the analyst (78%)
- Evade search robots/crawlers (only 3 kits)
- Check itself with various antiviruses (None)

Defensive component

What possibly can it be

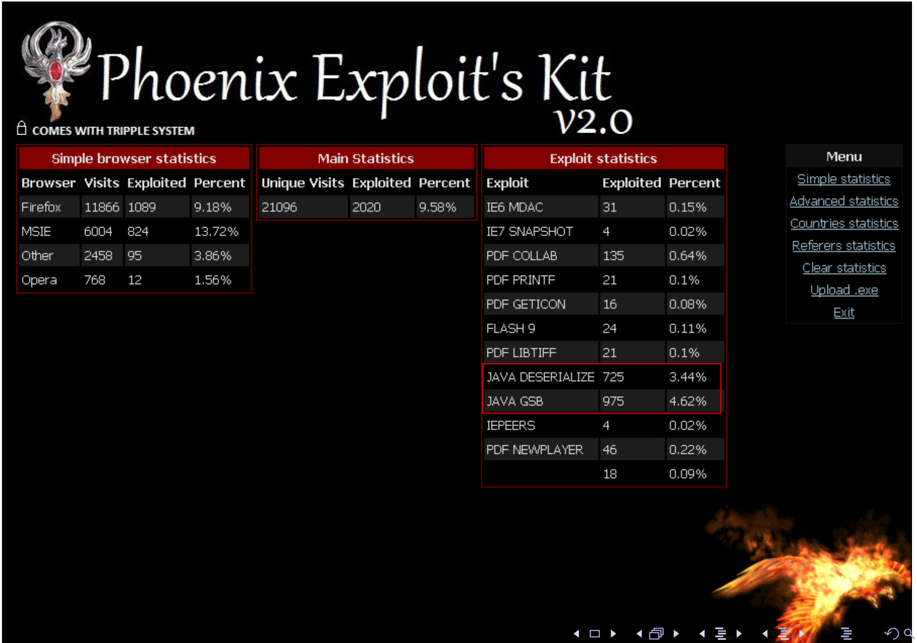
- Trick antivirus signatures using obfuscation (82%)
- Block repeating IP to prevent probing by the analyst (78%)
- Evade search robots/crawlers (only 3 kits)
- Check itself with various antiviruses (None)

Vienn diagram of defensive capabilites



Management component

Example



Phoenix Exploit's Kit v2.0

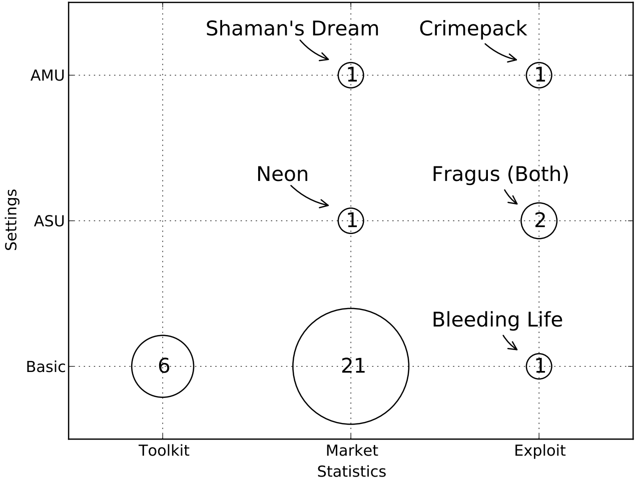
COMES WITH TRIPPLE SYSTEM

Simple browser statistics			
Browser	Visits	Exploited	Percent
Firefox	11866	1089	9.18%
MSIE	6004	824	13.72%
Other	2458	95	3.86%
Opera	768	12	1.56%

Main Statistics		
Unique Visits	Exploited	Percent
21096	2020	9.58%

Exploit statistics		
Exploit	Exploited	Percent
IE6 MDAC	31	0.15%
IE7 SNAPSHOT	4	0.02%
PDF COLLAB	135	0.64%
PDF PRINTF	21	0.1%
PDF GETICON	16	0.08%
FLASH 9	24	0.11%
PDF LIBTIFF	21	0.1%
JAVA DESERIALIZE	725	3.44%
JAVA GSB	975	4.62%
IEPEERS	4	0.02%
PDF NEWPLAYER	46	0.22%
	18	0.09%

Management component



The majority of the exploit kits (21) offer basic configuration options but report statistics for the owner to work with other markets.

Source code

The technology

All the kits analyzed are built upon PHP + MySQL

Source code

The technology

All the kits analyzed are built upon PHP + MySQL

Code protection

- Several kits use commercial code protection
- Few kits have ad-hoc (and very weak) protection
- Vast majority of the kits does not have any defenses

Source code

The technology

All the kits analyzed are built upon PHP + MySQL

Code protection

- Several kits use commercial code protection
- Few kits have ad-hoc (and very weak) protection
- Vast majority of the kits does not have any defenses

Code re-use

- GeoIP and PluginDetect are frequently used pieces of code
- 5 different kits share the same obfuscation routine
- Code repeat rate is 4%

Source code

The technology

All the kits analyzed are built upon PHP + MySQL

Code protection

- Several kits use commercial code protection
- Few kits have ad-hoc (and very weak) protection
- Vast majority of the kits does not have any defenses

Code re-use

- GeoIP and PluginDetect are frequently used pieces of code
- 5 different kits share the same obfuscation routine
- Code repeat rate is 4%

Looks like there is no common code base and most of the kits were developed independently

Conclusions and future work

Conclusions

- Exploit kits from 2007 to 2012 were analyzed: 5 years - same technology
- Very few vulnerabilities are exploited
- Exploits in the kits are quite outdated
- Profit is expected to come with large volumes of traffic rather than sophisticated attacks

Conclusions and future work

Conclusions

- Exploit kits from 2007 to 2012 were analyzed: 5 years - same technology
- Very few vulnerabilities are exploited
- Exploits in the kits are quite outdated
- Profit is expected to come with large volumes of traffic rather than sophisticated attacks

Future work

- What is the real success rate of an exploit kit?
- Can we measure a quality of the particular specimen?
- To answer these question we run a set of experiments in the virtualized environment.

That’s all, folks

Thank you for your attention!