Tracing Cyber Attacks...

...with Honey System

MTech Thesis Defence

By: Rohit Sehgal
Acknowledgement

Dr Sandeep Shukla

Committee members

Dr Manindra Agrawal
Dr Mainak Chaudhuri
Dr Tanaya Guha
Outline:

Honesystem Models:

• Methodology of Honeypots
• Types of Honeypots.
• Overview of Clientpot.
• Honeytokens & Gooligan

• Server Honeypots:
  1. HoneyFARM
  2. HoneyWEB
  3. HoneyFTP

* And analysis of attacks captured by these Honeypots.
Different threats:

1. Web Server is Malicious.
2. Attackers scanning for systems with open ports and exploiting services to gain access to machines.
3. Malware compromising Mobile device.
Honeysystems

- Honeytokens
- Honeyclient
- Server Honeypots
1. Honeypots

What are they & Why do we need them?
A honeypot is security resource whose value lies in being probed, attacked, or compromised.

What do we understand by this?

Lance Spitzner
Attacking Scenario:

- Attacker
- Benign Server
- Internet
- LAN

What if again attacked?

No more interesting ...

No Traces of Scripts

How can we inform others?
How Honeypots Help?

- Attacker
- Benign Server
- Internet
- LAN
- Honeypots
  - More attractive than others
  - Make difficult to reach
- Open ports

Honeypots make attacking more visible than others by making them more attractive and difficult to reach.
How?

Visibility:
1. Increase exposed ports
2. Easily crackable passwords
3. Allow easy fingerprinting

Attractive:
1. Offers more services
2. Good number of resources
2. Clientpot - A HoneyClient

Identifying Malicious Web servers
Clientpot

- Identify malicious web servers.

- Simple Light-weight honeyclient for Linux.

- Model uses software based virtualisation technique.

Nishit Majithia will be presenting most part of this work.
3. HoneyToken & Gooligan

Identifying “Gooligan” breached devices and more...
HoneyTokens

• Captures **Insider attacks** and much more ...

• **Password cracking** identifiable....

• Tokens may moves attackers to Honeypots ...

Unlike Other Honeypots can be deployed without much efforts
System under Attack

Hashed password file

Try to login using
unhashed passwords

Unhash the
passwords

Hash of all passwords

Compromised!!

Friend/Insider

root:root
admin:admin
...

root:xxxx
admin:yyyy
...
Why HoneyTokens?

Can we identify such in-genuine access that originated with illegal access of hashed password file?

Try to login using Hashed password file

Unhash the passwords

Alarm ... !!
Password File Compromised

Friend/Insider

root:root
admin:admin
...
root:toor
What is Gooligan Malware?

• Malware for Android devices.

• Steal authentication tokens.
Possible working of Gooligan Malware

Photo credits: http://blog.checkpoint.com/2016/11/30/1-million-google-accounts-breached-gooligan/
What is CVE-2016-5195?

• Privilege escalation bug.

• Allows users to **write on read only files**.

• Existed in Linux kernel version **2.6.22 - 4.2.0**.

• Patched in November, **2016**.
Breaking Android Sandbox Using CVE 2016-5195

- From Android version **Kitkat**, only few programs were allowed `setuid()`.
- Lollipop implemented **SELinux Policy**
- A few programs are still able to execute `setuid()`.
- `run-as, wpa_supplicant, fsck_disk` etc. Very limited in Count.
Let’s Understand it!!

• `wpa_supplicant` binary in `/system/bin/` is able to execute `setuid()`.
• We can’t change the working of this binary to become `root` ........
  i.e can’t execute `setuid(0,0,0)`
• Can we use CVE 2016-5195 to write to new `wpa_supplicant`?
• Newer version of `wpa_supplicant` can make us `root` ..!
Can we do some Mischief with wpa_supplicant?

- Modified it allow us to fetch the user id of the process to the desired one.
- `system` userid have access to the data in `/data/system/user/0/accounts.db` which stores the OAuth tokens in one file.
- We can attain `system` user id. And can fetch tokens.
/data/system/user/0/accounts.db from device
Any thing else we can do?

- **system** user have access to the file `/data/system/packages.list`.
- This file stores the mapping of each application with its **user-id**.
- E.g user-id of WhatsApp application on was **10089**, can directly use **wpa_supplicant** to become WhatsApp user.
- And can access the WhatsApp directory @ `/data/data/com.whatsapp/databases/msgstore.db`
WhatsApp Data compromised!!

WhatsApp stores the data in unencrypted form!!!
Back to Gooligan!

- It may have exploited CVE 2016-5195
- To fetch temporal system user.
- OAuth token can be used by Attacker to authenticate access to GApps.
HoneyTokens can save..... Post exploitation Idea

- Adding fake entry in `accounts.db` with an app which is not installed in the system.
- If attacker uses that fake token he will be trapped....
- Some one have accessed a `accounts.db` file.
- For an installed application, create a another `fake` entry.... chances to catch the attacker with one entry is 50%.
What needs to be done?

- Google has to keep track which account on which device installed which one of the GApp.

- Access to the fake token can generate alarm.
4. Service Honeypots

Capturing attackers ...
Deployed HoneyNetwork

Honey network composed of these many Honeypots created by all of us.

** Thanks to Nishit Majithia & Krishnaprasad P
HoneyFARM, HoneyFTP & HoneyWEB

Capturing attacks on SSH protocol, FTP protocol & WEB server
4.1 HoneyFARM

How do attackers target SSH protocol? How can we keep them busy for long?
HoneyFARM

- Model to Capture attacks on SSH protocol.
- Keep attacker busy for long in the system.
- Setups SSH redirection systems, that makes manual attacker loops around in few systems.
Model insights:

- Never let the manual attacker go out of the system.
- The entire system is built behind open source ssh proxy.
- Logs are maintained in the main physical system.
- The firewall rules prevent the machine from abusing other machines.
- The copy of the scripts used by the attacker is saved in main machines.
What else?

Binaries of modified ssh client used by attacker will be saved.

If attackers do use sftp to put malicious scripts, it will be captured.
4.2 HoneyFTP

How do attackers target FTP protocol? How do they use FTP to attack other?
HoneyFTP
Model insights:

- Logs are maintained in physical machine.
- FTP service runs in docker guest to ensure isolation.
- Log and Directory that FTP serves is maintained in docker host (Physical Machine).
- Main FTP docker is attached to few other docker machine to capture attacks “FTP bounce attack”
How do attackers attack web servers running in Public IPs?
HoneyWEB
Model insights:

- Docker compose environment that exposes port of proxy to the attacker.
- Proxy maintains the logs in the physical system.
- Server do not exposes vulnerability but static server the website in the folder from Physical machine mounted on the docker host.
5.1 Analysis of Attacks on HoneyFARM, HoneyFTP & HoneyWEB

..... with IP targeting Darknets

Capturing attacks on SSH protocol, FTP protocol & WEB server
5.1 Comparison between IP's targeting darknet & IP's targeting Honeypots

What is the pattern that can be extracted??
Common IP Address:

Map plotted using plot.ly
IP targeting darknet also targeted honeypots?

#137 in total

- SSH: 50
- WEB: 32
- Database: 30
- Telnet: 13
- FTP: 5
- SQLi: 3
- SMB: 4

*The traffic targeting darknet captured by Devashish & Nikhil in their UGP.*
5.4 HoneyFARM

Behaviour of attackers on SSH protocol.
Country wise attacks on SSH protocol

Majority of the attacks on SSH protocol were from Vietnam.

Probable reason could be the open proxies and VPNs.

The Attacks from Vietnam were to that of other countries.
Login Attempt count By IP s:

<table>
<thead>
<tr>
<th>Number of attempts</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attempts &lt;1000</td>
<td>1215</td>
</tr>
<tr>
<td>1000 ≤ Attempts &lt;10000</td>
<td>24</td>
</tr>
<tr>
<td>Attempts ≥ 10000</td>
<td>7</td>
</tr>
</tbody>
</table>
**IPs making attempt >= 10000**

First 3 and last 2 IPs belonged to the same network and were the part of **AS4134 CHINANET-BACKBONE** organization.

Tried same user name password combinations. May be have used same password and username files.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.177.172.24</td>
<td>12329</td>
</tr>
<tr>
<td>61.177.172.17</td>
<td>12680</td>
</tr>
<tr>
<td>61.177.172.55</td>
<td>15734</td>
</tr>
<tr>
<td>183.214.141.104</td>
<td>27170</td>
</tr>
<tr>
<td>116.31.116.10</td>
<td>38128</td>
</tr>
<tr>
<td>58.218.211.78</td>
<td>49315</td>
</tr>
<tr>
<td>58.218.204.223</td>
<td>71158</td>
</tr>
</tbody>
</table>

Attack may have compromised machine in same network.
SSH Attacks on different Locations beside NKN network.

Total unique IPs captured by them was **248**

80% of attacks were for Indian Digital Ocean Server.

Indian is at 3rd place in terms of Inviting Cyber Attacks
Binaries captured by HoneyFARM

# 88 in total

- XOR DDoS: 26
- Banwidth check: 13
- DoS: 13
- Full DoS Kit: 26
- Mirai: 20
- Backdoor: 1
- Trojan Ramgo: 1
- Dridex: 2
- Data file: 3

Total: 88
SSH Attack types:

- Terminal: 67
- Scripted: 115
- SFTP: 10

The attack trend for the last 10 days of HoneyFARM deployment.
IP address common amongst Various HoneyFARM deployment
Different HoneyFARM Deployment Scenarios

1. **Admin User + No Firewall Restriction** = **No Scripts Capture**
2. **Privileged Admin User + No Firewall Restriction** = **Few Scripts**
3. **Root User + Strict Firewall policies** = **Few scripts captured**
4. **Root User + No Firewall** = **Good amount of Scripts were captured** + **Cases of Exploitation of Honeypots**
Honeypot recreation time to Minimise its abusing

Scene 1
Scene 2
Scene 3
Scene 4
Scene 5
Scene 6

HoneyFARM recreation time
Numbe of Times Machine Compromised
Number of Scripts fetched

There is common pattern that attacker use

Compromise was identified using iftop
5.3 HoneyFTP

How FTP was abused to attack others?
Attacks on FTP server from Different Countries

- India: 40%
- United States: 13.33%
- Uzbekistan: 8.58%
- China: 7.1%
- Korea: 6.0%
- Nigeria: 5%
- Singapore: 6.66%

Attacked by 460 unique IPs.
Per day attack analysis on FTP

These peaks defining the “FTP Bounce Attack” on HoneyFTP.

Internal network that were scanned by the attacker were 10.0.0.0/8, 172.17.0.0/16 and 192.168.0.0/16 for the port 80, 22.

Intentionally made FTP Docker connected with others to capture ssh attack on that.

No attacks on it.
5.6 HoneyWEB

Similarities of attacks on HoneyWEB & HoneySQLi.
Country Wise attacks on HoneyWEB

Attacked by 568 unique IPs.

Attacking countries almost similar to HoneySQLi
Attack trends on daily basis:

The graph has similar pattern as to that of Honey WEB-SQLi**

**Deployed by Nishit Majithia
Attacks from Compromised machines

1. One IP of the Microsoft CERT organisation has attacked like:
   a. Scan all version of PHP and tried to get access to the files setup.php
   b. Tried to exploit the Linksys router vulnerability.
   c. Identify if the website is built using the word press framework to get access of setup.php and db.php
   d. Scan for pages like //w0Otw0Ot.at.ISC.SANS.DFind:
       //w0Otw0Ot.at.blackhats.romanian.anti-sec:}
Attacks from Compromised Devices:

1. Tried to exploit php exec() function downloaded “mirai” binaries for linux.
2. Attacks were from Jogree compromised devices or systems which made access to around 336 files access of the web server.
3. More than 30 IP address many times tried to exploit the vulnerability of struts framework by accessing login.action, test.action, *.action files.
Conclusion:

1. Different Honeypots were deployed for ~600 hrs, captured good number of malware.

2. Most of the times Attacker tried exploiting different framework’s Vulnerability.

3. Attackers do not waste their time in un-exploitable resources, dynamic honeypots can be framed to capture good attacks.
Thank you .... Questions ?