Introduction into Threat Hunting

$WHOAMI

- **Rasmus Männa**
- SecOps engineer in Transferwise
- Previously Cyber Security Expert in CERT-EE
- Long Telecom experience from VoIP and Networking with all the bells and whistles from regular network attacks up until Fraud
- all the way Blue Teamer
- https://github.com/razuz / https://twitter.com/razumlwr

$WHOAMI

- **Andres Elliku**
- SecOps engineer at TransferWise
- Previously Cyber Security Expert at CERT-EE
- Sys Admin before that, mostly focusing on Windows
- Occasional client-side Red Teamer at various exercises
- https://github.com/haam3r
- https://twitter.com/haam3r

TL;DR

- We are first and foremost technical people
- We’re not talking about a specific framework/standard/guide in this workshop
- We’re purely talking about our experiences and approaches
- All of the tools we use or mention in this workshop are open-source/free and they scale to practically any organisation
- Computers are hard … securing them even harder ...
- Hunting in a large infrastructure is a lot of work …
- With all the methods we cover their deeper background and suitable use cases

Ask questions !

Agenda

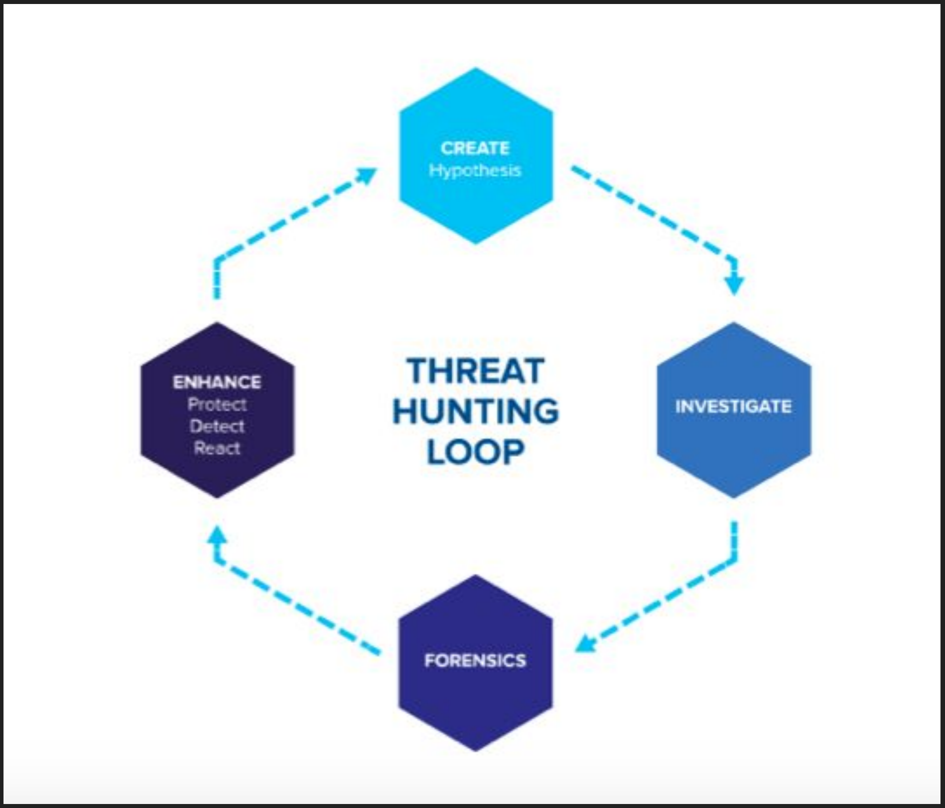
1. Introduction to Threat hunting
2. DNS Fuzzing
3. PassiveDNS
4. Certificate Transparency Log
5. Hunting in web server logs
6. Hunting with e-mails
7. Hunting in Windows logs
8. Wrapping it all up

What is Threat Hunting

What is Threat Hunting

Cyber threat hunting is an active cyber defence activity. It is **"the process** of proactively and iteratively searching through networks to detect and isolate advanced threats that evade existing security solutions." This is in contrast to traditional threat management measures, such as firewalls, intrusion detection systems (IDS), malware sandbox (computer security) and SIEM systems, which typically involve an investigation of evidence-based data **after** there has been a warning of a potential threat.

What is Threat Hunting



Why is Threat Hunting important

- **Assume breach** mentality
- Finding **skilled attackers** who have gotten past your (automated) **defenses**
- **Continual improvement** of your (automated) **detection capabilities**
- You cannot have **oversight** of everything that’s happening
- Supports **faster detection** of potential **compromise**
- Better **awareness** of your environment and attack surface
- **A lot** of accidental discoveries of **policy violations**
- Driver for **better data collection**

Types of Threat Hunting

IoC based threat hunting

- Is this indicator present or seen in my environment ?
- IoC-s can be collected from different sources:
  - Twitter
  - Partners (for example MISP)
  - Paid feeds

https://www.twitter.com/threathunting_

Hypothesis based threat hunting

- Can I find a specific malicious activity being used against my organisation ?
- Kind of goes hand-in-hand with IoC based hunting
- Can be based on MITRE ATT&CK framework

Baseline based threat hunting

- If I know the current state of my environment, can I detect something I haven’t seen before ?
- Expects larger set of data available about your infra
- Triggers lot of False Positives while Baseline is constantly shifting
- Using messaging software (eg Kafka) helps a lot
- Very effective to spot changes in your infra

Anomaly based threat hunting

- Sifting through the log data available to me, to spot irregularities that might be malicious
- Additionally applying patterns on your infra
- Useful in Fraud detection
- Standard behaviour is measured

Workshop resources

- You can either use the VirtualBox image in a local VM or ask for a VM
- Additionally we will use a tool called HELK (More on this later)
    - HELK Kibana instance is located at: https://helk.devspree.com
    - Credentials for connecting are:
        - user: helk
        - password: hunting

Threat Hunting Methods

https://www.twitter.com/threathunting_

DNS Fuzzing

What is DNS Fuzzing

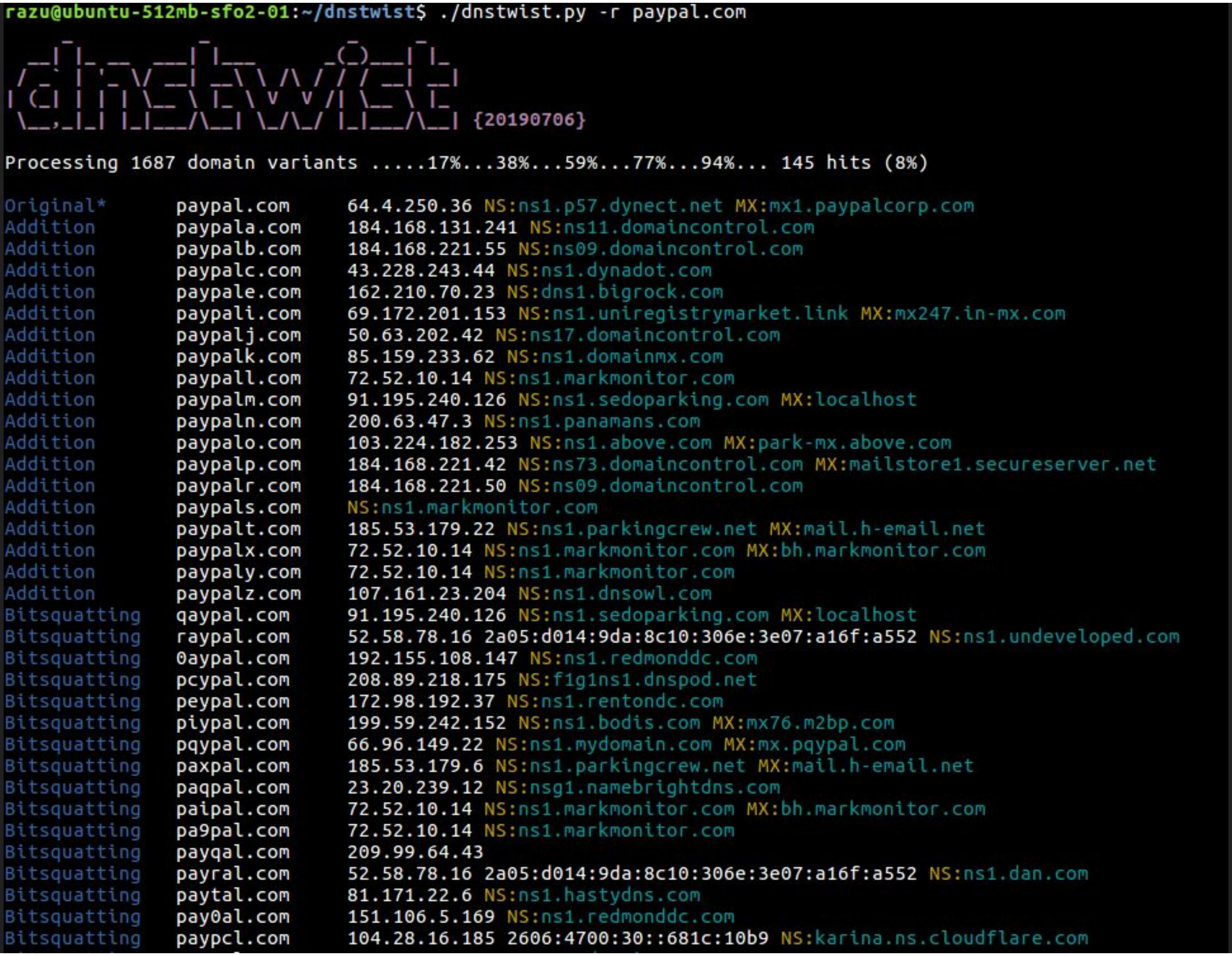
- An automated workflow for discovering potentially malicious domains related/targeting your organisation.
- Works by generating a large list of permutations based on a domain you provide and then checking if any of those permutations are in use
- These malicious domains might include things like typo squatting, unicode domains and suspicious sub domains

Benefits of DNS Fuzzing

- Provides another source to discover potentially malicious sites when they are booted up
- Find domains when actors register them i.e. find potentially malicious domains when actors start using them
- Keep state of what is out there

Tool - DNSTwist

- https://github.com/elceef/dnstwist
- The idea is quite straightforward: dnstwist takes in your domain name as a seed, generates a list of potential phishing domains and then checks to see if they are registered. Additionally it can test if the mail server from MX record can be used to intercept misdirected corporate e-mails and it can generate fuzzy hashes of the web pages to see if they are live phishing sites.



```
razu@ubuntu-512mb-sfo2-01:~/dnstwist$ ./dnstwist.py -r paypal.com
     _           _            _     _
  __| |_ __  ___| |___      _(_)___| |_
 / _` | '_ \/ __| __\ \ /\ / / / __| __|
| (_| | | | \__ \ |_ \ V  V /| \__ \ |_
 \__,_|_| |_|___/\__| \_/\_/ |_|___/\__| {20190706}

Processing 1687 domain variants .....17%...38%...59%...77%...94%... 145 hits (8%)

Original*      paypal.com     64.4.250.36 NS:ns1.p57.dynect.net MX:mx1.paypalcorp.com
Addition       paypala.com    184.168.131.241 NS:ns11.domaincontrol.com
Addition       paypalb.com    184.168.221.55 NS:ns09.domaincontrol.com
Addition       paypalc.com    43.228.243.44 NS:ns1.dynadot.com
Addition       paypale.com    162.210.70.23 NS:dns1.bigrock.com
Addition       paypali.com    69.172.201.153 NS:ns1.uniregistrymarket.link MX:mx247.in-mx.com
Addition       paypalj.com    50.63.202.42 NS:ns17.domaincontrol.com
Addition       paypalk.com    85.159.233.62 NS:ns1.domainmx.com
Addition       paypall.com    72.52.10.14 NS:ns1.markmonitor.com
Addition       paypalm.com    91.195.240.126 NS:ns1.sedoparking.com MX:localhost
Addition       paypaln.com    200.63.47.3 NS:ns1.panamans.com
Addition       paypalo.com    103.224.182.253 NS:ns1.above.com MX:park-mx.above.com
Addition       paypalp.com    184.168.221.42 NS:ns73.domaincontrol.com MX:mailstore1.secureserver.net
Addition       paypalr.com    184.168.221.50 NS:ns09.domaincontrol.com
Addition       paypals.com    NS:ns1.markmonitor.com
Addition       paypalt.com    185.53.179.22 NS:ns1.parkingcrew.net MX:mail.h-email.net
Addition       paypalx.com    72.52.10.14 NS:ns1.markmonitor.com MX:bh.markmonitor.com
Addition       paypaly.com    72.52.10.14 NS:ns1.markmonitor.com
Addition       paypalz.com    107.161.23.204 NS:ns1.dnsowl.com
Bitsquatting   qaypal.com     91.195.240.126 NS:ns1.sedoparking.com MX:localhost
Bitsquatting   raypal.com     52.58.78.16 2a05:d014:9da:8c10:306e:3e07:a16f:a552 NS:ns1.undeveloped.com
Bitsquatting   0aypal.com     192.155.108.147 NS:ns1.redmonddc.com
Bitsquatting   pcypal.com     208.89.218.175 NS:f1g1ns1.dnspod.net
Bitsquatting   peypal.com     172.98.192.37 NS:ns1.rentondc.com
Bitsquatting   piypal.com     199.59.242.152 NS:ns1.bodis.com MX:mx76.m2bp.com
Bitsquatting   pqypal.com     66.96.149.22 NS:ns1.mydomain.com MX:mx.pqypal.com
Bitsquatting   paxpal.com     185.53.179.6 NS:ns1.parkingcrew.net MX:mail.h-email.net
Bitsquatting   paqpal.com     23.20.239.12 NS:nsg1.namebrightdns.com
Bitsquatting   paipal.com     72.52.10.14 NS:ns1.markmonitor.com MX:bh.markmonitor.com
Bitsquatting   pa9pal.com     72.52.10.14 NS:ns1.markmonitor.com
Bitsquatting   payqal.com     209.99.64.43
Bitsquatting   payral.com     52.58.78.16 2a05:d014:9da:8c10:306e:3e07:a16f:a552 NS:ns1.dan.com
Bitsquatting   paytal.com     81.171.22.6 NS:ns1.hastydns.com
Bitsquatting   pay0al.com     151.106.5.169 NS:ns1.redmonddc.com
Bitsquatting   paypcl.com     104.28.16.185 2606:4700:30::681c:10b9 NS:karina.ns.cloudflare.com
```

Task 1: Running DNSTwist

1. Install DNSTwist
   a. sudo apt-get install python3-dnspython python3-geoip python3-whois python3-requests python3-ssdeep python3-pip automake libtool
   b. git clone https://github.com/elceef/dnstwist.git
   c. cd dnstwist
2. Run DNSTwist for the domain that you would like to analyse:
   a. ./dnstwist.py -r domain.name

PassiveDNS

What is PassiveDNS

- Passive monitoring and logging of all DNS queries and responses from monitored network traffic
- Requires network span to be available for monitoring (can co-exist with IDS infra)

Benefits of PassiveDNS

- Gives overview of domains contacted from within your perimeter
- Well implemented setup also answers who contacted it
- Great tool for discovering DNS tunneling

Tools

- Gamelinux passivedns - https://github.com/gamelinux/passivedns
- Suricata DNS log - https://github.com/OISF/suricata/blob/master/suricata.yaml.in#L201-L230
- BRO/Zeek - https://docs.zeek.org/en/stable/scripts/base/protocols/dns/main.zeek.html
- Farsight Security sensor - https://github.com/farsightsec/sie-dns-sensor

Public databases

- CIRCL passivedns - https://www.circl.lu/services/passive-dns/
- Farsight Security - https://www.farsightsecurity.com/solutions/dnsdb/
- ….

Example log

```
#timestamp||dns-client ||dns-server||RR class||Query||Query Type||Answer||TTL||Count
1322849924.408856||10.1.1.1||8.8.8.8||IN||upload.youtube.com.||A||74.125.43.117||46587||5
1322849924.408857||10.1.1.1||8.8.8.8||IN||upload.youtube.com.||A||74.125.43.116||420509||5
1322849924.408858||10.1.1.1||8.8.8.8||IN|| www.adobe.com.||CNAME||www.wip4.adobe.com.||43200||8
1322849924.408859||10.1.1.1||8.8.8.8||IN|| www.adobe.com.||A||193.104.215.61||43200||8
1322849924.408860||10.1.1.1||8.8.8.8||IN||i1.ytimg.com.||CNAME||ytimg.l.google.com.||43200||3
1322849924.408861||10.1.1.1||8.8.8.8||IN||clients1.google.com.||A||173.194.32.3||43200||2
```

Task 2: How many IP-s behind a domain

- Find out to how many IPs the domain **cert.gov.lk** point to

http://192.168.12.20/hunting/

http://138.197.214.229/training/

```
tar xvzf passivedns…..
```

Task 3: Find a suspicious domain from logs

- You can try different methods here
    - Look for suspicious top level domains
    - long domain names
    - multiple levels

Task 4: Find a DNS beacon

- Find a PassiveDNS log entry that indicates the use of A record based DNS tunneling
- Other types of DNS tunneling:
  - TXT record based tunneling
  - HTTP based tunneling

Note that beacon is not CNAME (needed to hide it)

Certificate Transparency Log

https://www.twitter.com/threathunting_

What is Certificate Transparency Log

A newer feature of the PKI ecosystem, where the issuance of certificates is logged into a public logstream

As an example, every time a Let’s Encrypt certificate is issued for a website, a publicly available log entry is created detailing the certificates contents

This provides us with yet another source to discover malicious sites

Benefits of Certificate Transparency Log

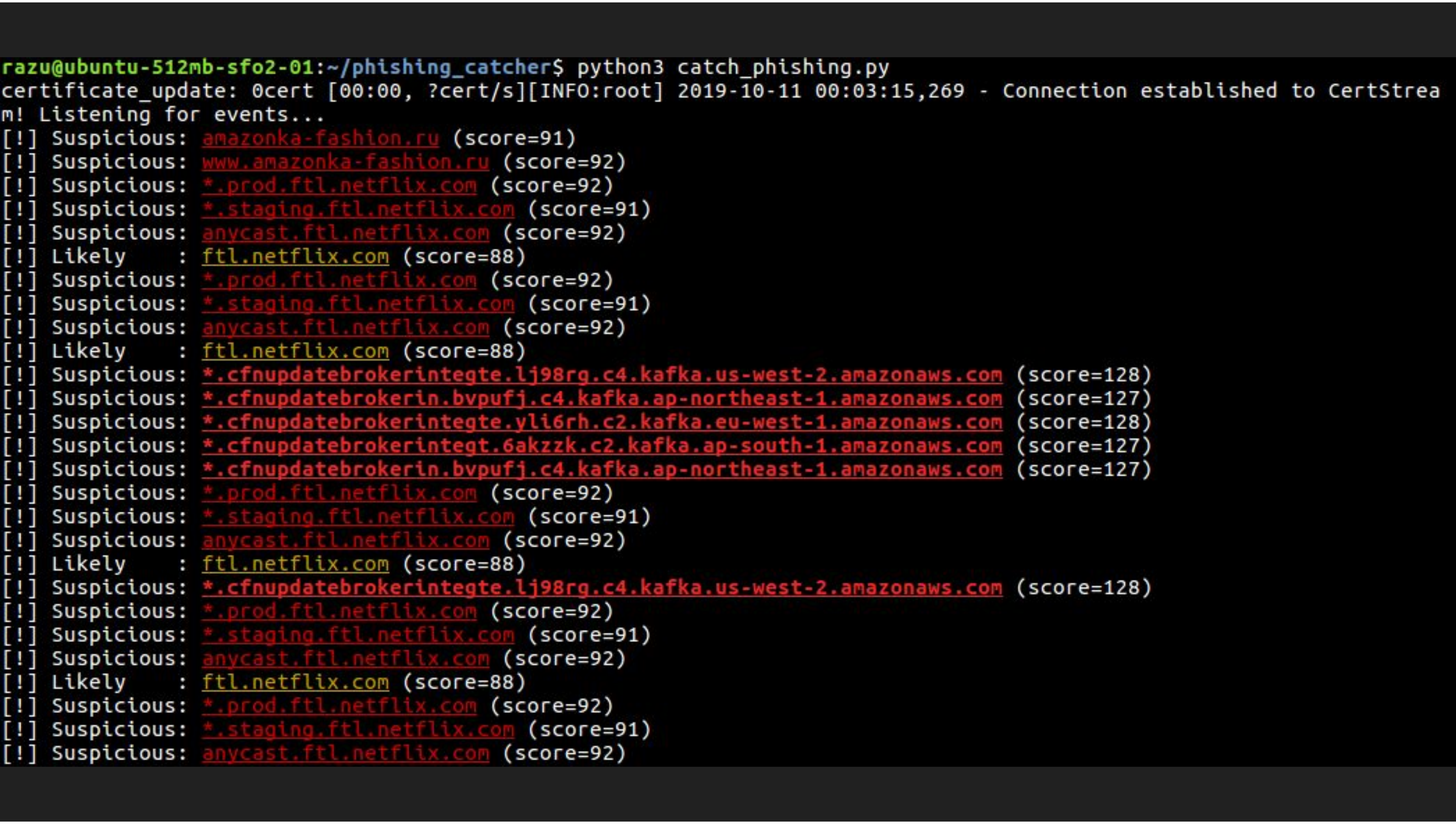
- Provides another source to discover potentially malicious sites when they are booted up
- Gives capability to monitor even entire TLDs
- Find domains while actors are working on them

Tool - Phishing Catcher

- http://github.com/x0rz/phishing_catcher
- Phishing catcher will monitor your provided list of domains and triggers an alert if certificate transparency log entry similarity score goes above defined threshold.
- This is done via scoring following parameters:
  - TLDs - potentially malicious TLDs
  - High entropy
  - lookalike characters - for example 1 vs l
  - Levenstein distance - for example paypal vs paypol
  - occurence or “-” - for example www-paypal-com-index-php.malware.net
  - deeply nested domains - for example www.paypal.com.this.is.my.awesome.name.com
  - etc ...

Tool - Phishing Catcher

- Tool can be highly customised and new scoring sets added
- DNS fuzzing can be one of the inputs for Certificate Transparency Log monitoring



```
razu@ubuntu-512mb-sfo2-01:~/phishing_catcher$ python3 catch_phishing.py
certificate_update: 0cert [00:00, ?cert/s][INFO:root] 2019-10-11 00:03:15,269 - Connection established to CertStrea
m! Listening for events...
[!] Suspicious: amazonka-fashion.ru (score=91)
[!] Suspicious: www.amazonka-fashion.ru (score=92)
[!] Suspicious: *.prod.ftl.netflix.com (score=92)
[!] Suspicious: *.staging.ftl.netflix.com (score=91)
[!] Suspicious: anycast.ftl.netflix.com (score=92)
[!] Likely    : ftl.netflix.com (score=88)
[!] Suspicious: *.prod.ftl.netflix.com (score=92)
[!] Suspicious: *.staging.ftl.netflix.com (score=91)
[!] Suspicious: anycast.ftl.netflix.com (score=92)
[!] Likely    : ftl.netflix.com (score=88)
[!] Suspicious: *.cfnupdatebrokerintegte.lj98rg.c4.kafka.us-west-2.amazonaws.com (score=128)
[!] Suspicious: *.cfnupdatebrokerin.bvpufj.c4.kafka.ap-northeast-1.amazonaws.com (score=127)
[!] Suspicious: *.cfnupdatebrokerintegte.yli6rh.c2.kafka.eu-west-1.amazonaws.com (score=128)
[!] Suspicious: *.cfnupdatebrokerintegt.6akzzk.c2.kafka.ap-south-1.amazonaws.com (score=127)
[!] Suspicious: *.cfnupdatebrokerin.bvpufj.c4.kafka.ap-northeast-1.amazonaws.com (score=127)
[!] Suspicious: *.prod.ftl.netflix.com (score=92)
[!] Suspicious: *.staging.ftl.netflix.com (score=91)
[!] Suspicious: anycast.ftl.netflix.com (score=92)
[!] Likely    : ftl.netflix.com (score=88)
[!] Suspicious: *.cfnupdatebrokerintegte.lj98rg.c4.kafka.us-west-2.amazonaws.com (score=128)
[!] Suspicious: *.prod.ftl.netflix.com (score=92)
[!] Suspicious: *.staging.ftl.netflix.com (score=91)
[!] Suspicious: anycast.ftl.netflix.com (score=92)
[!] Likely    : ftl.netflix.com (score=88)
[!] Suspicious: *.prod.ftl.netflix.com (score=92)
[!] Suspicious: *.staging.ftl.netflix.com (score=91)
[!] Suspicious: anycast.ftl.netflix.com (score=92)
```

Task 5: Monitor CT log for a specific domain

- Install phishing_catcher
  - git clone https://github.com/x0rz/phishing_catcher.git
  - cd phishing_catcher
  - pip3 install -r requirements.txt
  - python3 catch_phishing.py
- Modify the config/code so that it would also catch ***.lk**

Hunting in web server logs

What are webserver logs

Webserver logs is a valuable data source (which is often overlooked) which provides info of:

- who is accessing your website(s)
- what are they doing there

Provides you following hunting data (even honeypots) :

- user-agents
- URI paths
- referrers
- POST/GET/… request size

Benefits from webserver logs

- unusual user-agents
- lazy attackers (for example http://burp/)
- new exploitation techniques when attackers start using them
- automated scans or attacks (eg noise)
- external actors who are interested in you
- 3rd party referrals (might even be laptops eg actors working on you)

Tool

… Hands and eyes ...

Task 6: Find the last unique URI path

- Write some Python code to find the latest (based on timestamp) unique URI path in the logs

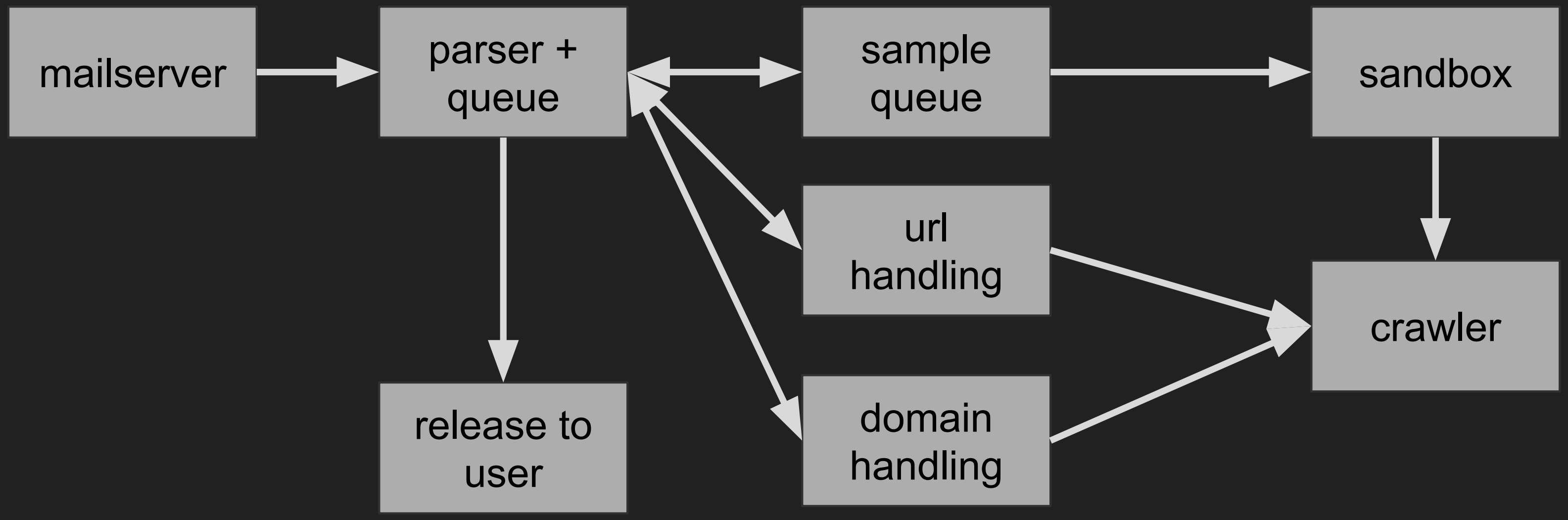
Hunting with e-mails

https://www.twitter.com/threathunting_

What does it mean ?

- E-mails/mailserver is highly valuable source of discovering attacks or attackers
- In there you can find:
  - malicious attachments
  - phishing links
- Complexity of hunting in the environment is considered as **very hard** (especially if you’re a financial institution … yes, macros)

Workflow


mailserver
parser + queue
sample queue
sandbox
url handling
crawler
domain handling
release to user

Tooling

- Postfix
- Custom queue scripts
- Sandbox (Cuckoo ?)
- Crawler software (usually custom)
- Yara rules
- ClamAV
- IP and domain filters/blacklists ….
- Ratelimiters

Hunting in Windows logs

Benefits of Windows logs

- Detect attacks against your endpoints
- Find policy violations of your users
- Help out IT by providing them an awesome way of debugging user problems

Types of Windows logs

- **Built-in Windows logs** - Examples are scheduled task creation, authentication events, RBAC modifications and much more
- **PowerShell scriptblock logging** - Logging of everything that is done on a Windows box using PowerShell (Based on ETW log traces)
- **Sysmon** - An awesome tool by Microsoft that extends the built-in log sources of Windows. Can log things like process creations, network connections, driver and image loading, DNS queries (uses ETW), registry and file system changes and much more. All of this with hashes and guid’s allowing to tie together events of different types
- **SilkETW** - Consumer for Event Tracing for Windows providers. Basically tap into the debug log stream of Windows that was originally meant for Microsoft internal developers. Allows for things like .NET introspection, logging kernel API calls

Consuming Windows logs

- Winlogbeat
- Windows Event Forwarding
- Agents for commercial SIEM-s, like the Splunk Universal Forwarder

HELK

- https://github.com/Cyb3rWard0g/HELK
- Open-source platform combining multiple open source tools to basically provide you a free SIEM
- Core toolset is ELK + Kafka
- Provides log ingestion, parsing, storage and analysis capabilities
- Alerting with Elastalert & Sigma
- Enrichment with Kafka
- Also has some built in Kibana dashboards for pretty pictures
- Jupyter notebooks and much more...

Task 7: Log into the Kibana dashboard of HELK

- Just to get started log into the Kibana console at:
  - http://helk.devspree.com
- Browse around, check out the different sections and take a look at the logs for a few minutes
- When you’re done, we can start hunting...

https://www.twitter.com/threathunting_

Task 8...n

- Now that you’ve had a chance to familiarize yourself with Kibana, we can start hunting
- We have an example dataset loaded into HELK. This dataset is correlated to the MITRE ATT&CK framework
- We’re gonna take a hypothesis based approach, to finding evil in these logs
- We’re gonna find different techniques from various tactics in the ATT&CK framework

Task 8: Inital compromise

- Hypothesis: An attacker might try to execute code using a scripting engine, to gain an initial foothold
- What was ran, to establish the first beacon?

Task 9: Inital Discovery

- Hypothesis: After gaining an initial foothold, an attacker will likely want to establish situational awareness
- What built-in utilities did they use?

Task 10: Discovery for Lateral Movement

- Hypothesis: An attacker might try and discover their options for lateral movement
- What kinds of things did the attacker check for?

Task 11: Privilege Escalation

- Hypothesis: An attacker will want to escalate their privileges on the host
- Can you find evidence for this?

Task 12: Credential Access

- Hypothesis: An attacker would want to collect credentials for lateral movement
- The attacker found a text file and looked at it’s content using PowerShell. What’s the name of the file?

Task 13: Lateral Movement

- Hypothesis: An attacker will want to move laterally
- Can you find evidence of the attacker laterally moving to the domain controller?

Task 14: Persistence

- Hypothesis: An attacker will want to maintain access through a persistence technique
- What utility was replaced to achieve persistence?

Task 15: Collection

- Hypothesis: An attacker will want to collect the data into one place, before exfiltrating it
- What actions were taken to do this? (Hint: There was PowerShell involved)

https://www.twitter.com/threathunting_

Task 16: Exfiltration

- Hypothesis: An attacker will want to exfiltrate stolen data
- What was used to do this? (Hint: The first thing was a renamed archive tool)

Task 17: Execution of Persistence

- Hypothesis: After losing their foothold, an attacker will want to utilize their previously installed persistence
- What command did the attacker run after regaining persistence?

Wrapping it all up

Collaborative tooling

MISP

- Malware Information Sharing Platform - https://www.misp-project.org/
- Good source of information for CERTs and IoC based hunting
- Enables you to share IoCs to other parties
- Very good tool to also manage IoCs internally
- (Be careful when you share data … False Positives are **not cool**)

The Hive

- Ticketing for Incident handlers - https://thehive-project.org/
- Case and IoC management which can be highly automated
- Correlation between cases is core functionality
- Flexible reporting capabilities for management
- Combined with Cortex provides automated enrichment and reactions
- Templating for predefined playbooks on how to handle specific types of incidents

IntelMQ

- Automated collection and parsing of Threat Intelligence feeds
- https://github.com/certtools/intelmq

SIGMA

- Open source repository of SIEM rules in a standardized format
- All rules in yaml format
- Lots of community input
- Provides multiple different backends for translating rules for specific platforms e.g. ELK, Splunk, QRadar, Graylog etc.
- https://github.com/Neo23x0/sigma

YARA

- The pattern matching swiss knife for malware researchers (and everyone else)
- https://virustotal.github.io/yara/
- https://github.com/InQuest/awesome-yara

SYSMON

- https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
- https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1554993664.pdf
- https://github.com/SwiftOnSecurity/sysmon-config
- https://github.com/ion-storm/sysmon-config
- https://blogs.technet.microsoft.com/motiba/2017/12/07/sysinternals-sysmon-suspicious-activity-guide/
- https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES
- https://github.com/olafhartong/sysmon-modular
- https://github.com/BlueTeamLabs/sentinel-attack

Various good resources/references

Good stuff on Windows forensiscs - http://www.hexacorn.com/blog/

Goldmine of Threat Hunting ideas and examples - https://twitter.com/SBousseaden

The dataset we used for the Windows tasks - https://github.com/hunters-forge/mordor

Good blogs to check out for threat hunting, red teaming etc:

- https://blog.harmj0y.net/
- https://posts.specterops.io
- http://www.exploit-monday.com/

References continued

- https://github.com/hunters-forge/ThreatHunter-Playbook - Playbooks for threat hunting
- https://github.com/clong/DetectionLab - Windows threat hunting lab/playground
-

Thank you

Questions?

https://www.twitter.com/threathunting_