Detecting Traffic to Recently Unparked Domains with Analysis Pipeline
Dan Ruef

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Copyright 2015 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon®, CERT® and FloCon® are registered marks of Carnegie Mellon University.

DM-0002981
Agenda

Define parked/unparked domains
Why are they interesting?
Use Pipeline to detect recently unparked domains
Detect traffic to IP addresses with recently unparked domains
Limit false positives
Results of testing on SEI network
Other DNS fun with Pipeline
Domain Parking / Unparking

For this presentation, a domain is considered **parked** when its associated IP is:

- 127.0.0.0/8
- 10.0.0.0/8
- 192.168.0.0/16
- 172.16.0.0/12
- 255.255.255.254/31
- 0.0.0.0
- 1.1.1.1

It becomes **unparked** when associated with a routable address.
Why can it be bad?

Changes in the control plane can be notable, even if not malicious

Some malicious uses of domain parking:

- Regulation of malware phoning home
  - Use parked IP address to keep malware quiet
  - Unpark IP address to receive data from malware
- Change ownership or location of existing malware
  - Redirect to the new owner of the implant
  - Redirect to newly hacked server after original one cleaned
What are we trying to find?

Domains that become unparked

Look for traffic of any sort being sent to IP addresses associated with those domains.
How do we do this?

Examine DNS A record response records
Look for responses with IP addresses in
   127.0.0.0/8, 10.0.0.0/8, 192.168.0.0/16,
   172.16.0.0/12, 255.255.255.254, 255.255.255.255, 0.0.0.0, 1.1.1.1
Build an IPSet with these addresses called: parked.set
Record the domain name
Look for records with those domain names with IP addresses that aren’t in the list above.
Record the unparked IP
Look for traffic sent from our network to that IP
Test Network setup

Multiple YAF sensors feed a super_mediator

Pipeline gets a copy of the full stream

Pipeline accept additional data sources
Pipeline handles DNS?

Pipeline version 4.x only handled SiLK flow records.
Pipeline version 5.x has expanded its input possibilities
  SiLK – just like v4
  YAF with full DPI (that’s how we get DNS)
  IPFIX – any raw ipfix can be ingested

Schemas are generated dynamically giving pipeline access to the fields without a priori knowledge of the record format

(Yes, YAF is IPFIX, but YAF data sources get special processing)

Pipeline 5.3 publicly released September 30th, 2015.
Find Parked Domains

FILTER parkedDomains
    rrIPv4 IN LIST "parked.set"
END FILTER

*parked.set consists of 127.0.0.0/8, 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 255.255.255.254, 255.255.255.255, 0.0.0.0, 1.1.1.1
Record Parked Domains

If a flow passes emptyDomainNames filter, record the dnsQName value in a list named domainsWithNoIP. Keep those values for 1 day.

INTERNAL FILTER park
  FILTER parkedDomains
dnsQName domainsWithNoIP 1 DAY
END INTERNAL FILTER
Find parked domains now unparked

FILTER unparked
dnsQName IN LIST domainsWithNoIP
rrIPv4 NOT IN LIST "parked.set"

END FILTER

*parked.set consists of 127.0.0.0/8, 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 255.255.255.254, 255.255.255.255, 0.0.0.0, 1.1.1.1
Record unparked IP Addresses for unparked domains

INTERNAL FILTER unpark
FILTER unparked
rrIPv4 unparkedDomainIPs 1 DAY
END INTERNAL FILTER
Our configuration so far

FILTER parkedDomains
  rrlIPv4 IN LIST “parked.set”
END FILTER

INTERNAL FILTER park
  FILTER parkedDomains
    dnsQName domainsWithNoIP
    1 DAY
END INTERNAL FILTER

FILTER unparked
  dnsQName IN LIST domainsWithNoIP
  rrlIPv4 NOT IN LIST “parked.set”
END FILTER

INTERNAL FILTER unpark
  FILTER unparked
    rrlIPv4 unparkedDomainIPs
    1 DAY
END INTERNAL FILTER
Find traffic to unparked domain IP addresses

FILTER trafficToUnparked
destinationIPv4Address IN LIST unparkedDomainIPs

END FILTER
Alert on traffic to unparked IP addresses

EVALUATION reportTrafficToUnparked
FILTER trafficToUnparked
ALERT ALWAYS
ALERT EVERYTHING
CHECK EVERYTHING PASSES
END CHECK
END EVALUATION
Results from live SEI* data

False positives at each step in analysis

Domain parking is not always malicious or even interesting

Let’s get rid of them

*Thanks Mike Pochan
False positive #1

Valid security sites use DNS to receive requests and return results

DNS query with request prepended to site’s domain:
• 1.2.3.4.securitysite.com
• potential.malicious.domain.securitysite.com

Replies are returned using responses in the 127.0.0.0/8 network
For example:
• 127.0.0.1 means OK
• 127.0.1.1 means malicious

These look like parked domains! Let’s not record these.
Filtering out security sites

Change filter that identifies parked domains

FILTER parkedDomains

  rrIPv4 IN LIST “parked.set”

  DNS_SLD(dnsQName) NOT IN LIST ["cmu", "cert",
                   "barracudacentral", "surriel", "spamhaus", "uribl", "isipp",
                   "root-servers", "dnswl", "sorbs", "senderscore", "support-
                    intelligence", "mcafee", "surbl", "nessus", "dynect",
                   "akadns", "quadranel"]

END FILTER
DNS Derived Fields

dnsQName is: a.B.domain.com.

DNS_PUBLIC_SUFFIX(dnsQName) = com
DNS_SLD(dnsQName) = domain
DNS_HOST(dnsQName) = a.B
DNS_PRIVATE_NAME(dnsQName) = a.B.domain
DNS_SLD+TLD(dnsQName) = domain.com
DNS_INVERT(dnsQName) = com.domain.B.a
DNS_NORMALIZE(dnsQName) = a.b.domain.com

(All lower case. Remove any starting or ending dots)
False positives #2: Ignore internal addresses

When looking for unparked domains, ignore those whose IP address is on the internal network.

FILTER unparked
  dnsQName IN LIST domainsWithNoIP
  rrIPv4 NOT IN LIST "parked.set", "internalSubnet.set"]
END FILTER
False positive #3: Safe* sites

Addresses within the IP spaces of Apple, Amazon, Google, and Microsoft.
External facing networks of CMU, SEI, and CERT.

FILTER unparked
dnsQName IN LIST domainsWithNoIP
rrIPv4 NOT IN LIST ["parked.set", "internalSubnet.set", "safeIPSpace.set"]

END FILTER

*I know safe is a dangerous word and cannot be assumed, but let’s pretend
False positive #4: Follow on DNS Traffic

When identifying traffic to unparked domain IP address, ignore traffic coming from our internal DNS servers

FILTER trafficToUnparked
    destinationIPv4Address IN LIST unparkedDomainIPs
    sourceIPv4Address NOT IN LIST "internalDNSServers.set"
END FILTER
FILTER parkedDomains
   rrlIPv4 IN LIST “parked.set”
END FILTER

INTERNAL FILTER park
   FILTER parkedDomains
   dnsQName domainsWithNoIP
   1 DAY
END INTERNAL FILTER

INTERNAL FILTER unpark
   FILTER unparked
   rrlIPv4 unparkedDomainIPs
   1 DAY
END INTERNAL FILTER

FILTER trafficToUnparked
   destinationIPv4Address IN LIST unparkedDomainIPs
END FILTER

EVALUATION reportTrafficToUnparked
   FILTER trafficToUnparked
   ALERT ALWAYS
   ALERT EVERYTHING
   CHECK EVERYTHING PASSES
END CHECK

END EVALUATION
See unparked domains in alerts

Add evaluation to alert when an unparked domain is discovered

Then we get the \{domain, IP\} tuple

Replace the “unpark” internal filter an evaluation alerting on unparked IP addresses.

Alerts will contain that:

• the IP
• the DNS flow record,
• the dnsQName as an extra field.

We now have a record of unparked domains and associated IPs
Replacement Evaluation

EVALUATION unparkedDNS_IP
  FILTER unparked
  FOREACH rrlIPv4
  EXTRA ALERT FIELD dnsQName
  CHECK THRESHOLD
    RECORD COUNT > 0
  END CHECK
  OUTPUT TIMEOUT 1 DAY
  OUTPUT LIST rrlIPv4 unparkedDomainIPs
END EVALUATION
Results

Three interesting domains found:
• x.bidswitch.net
  • On a list of phishing domains
• p.rfihub
  • Associated with a sinkhole IP
• ads.mp.mydas.mobi
  • Associated with a sinkhole IP
  • String found in 14 malware sample on Virus Total.
DNS Watchlisting

FILTER blacklist
dnsQName IN LIST “dnsBlacklist.txt”
END FILTER

EVALUATION badDNS
FILTER blacklist
ALERT ALWAYS
ALERT EVERYTHING
CHECK EVERYTHING PASSES
END CHECK
EXTRA ALERT FIELD dnsQName
END EVALUATION

dnsBlacklist.txt

##format:dns
inbox.google.com
maps.google.com
cmu.edu
sei.cert.org
pittsburgh.pirates.mlb.com
Pipeline Can Detect Fast Flux Networks

PMAP asn "pmaps/20151027.bgp.pmap"

EVALUATION FFv4
  ALERT ALWAYS
  ALERT EVERYTHING
  CHECK FAST FLUX
    IP_FIELD sourceIPv4Address 5
    ASN asn 5
    DNS dnsQName 5
    NODE MAXIMUM 150000
  END CHECK
END EVALUATION
SIE Data Processing

Security Information Exchange DNS data processing
  • CERT converts from nmessage to daily rollup CSV
  • Emily Sarneso used pyfixbuf to convert the record to IPFIX
  • Pipeline can process any IPFIX record

Searching for unparked domains from a 1 day rollup
  • 343,828,409 records
  • Found 25 unparked domains
  • 3 minutes 35 seconds
Thank you for listening

Dan Ruef
druef@cert.org
netsa-help@cert.org