TBD: To Block connection to malicious host by using “DQB” and "Shutdowner"

Kunio Miyamoto, Ph.D.
NTTDATA-CERT, NTT DATA Corporation
Kunio.Miyamoto@nttdata.com
Index

• Preliminary
• Decepting and Security live next door to each other
• Simple Specification/Architecture
• Simple Operation
• Jackpot!
• Conclusion
Preliminary: Our infrastructure Overview

• We use large amount of computers(above 20k~30k, Windows run most of computers)
  • Many of them: Windows Embedded(without MS17-010 patches)
• We have deployed and been operating Security Solutions as below:
  • Firewall(by security vendor)
  • Quarantine(Patch Management and Internet Access Control) (by security vendor)
  • USB port control (by security vendor)
  • URL filter (by security vendor)
  • End Point Security Software like virus scanner (by security vendor)
  • SIEM(by ourselves)
• Too many blackbox 😞
Preliminary: Responding to malicious URL

• Find suspicious URL by our SIEM or others
  • Consider whether URL is malicious or not(<1h)
  • Send URL Filter operators the request adding malicious URL(s) to URL filter
  • *(wait for a few hours)* (>2~3h)
    sometimes waits a few days(<1w)
  • Done

• If malicious URLs are found oftenly, URL Filter operators receive requests oftenly
  • Too heavy to process requests
Preliminary: Before DQB and Shutdowner

Problem 1: URL Filter operation process (adding malicious URL) needs more time (a few hours ~ a week).

Problem 2: Operation of the isolating infected host is the task of the host administrator.

NTTDATA-CERT

Malicious URL

Manual operation

IT manager

IR/Forensics
Decepting and Security live next door to each other

We Decept a Certain kind of Responses for Keep Security
Deception and Security

• Deception can make attackers to spend their time/costs for attacks more

• DQB and Shutdowner are system to decept
  • DQB(DNS Query Blocker): DNS response deception, don’t block DNS Query 😊
  • Shutdowner: TCP response deception

• Deploying these systems to:
  • DQB: same segment that the cache DNS in NTT DATA is placed to decept efficiently
  • Shutdowner: same segment of the Proxy Load Balancer to stop C&C communication
Simple Specification/Architecture

Complex Specification/Architecture makes work slower 😞
Basic Idea is a very simple

• If ②’ is received by Victim faster than ②, ② from Attacker is ignored
The ID field identifies the query and is echoed in the response so they can be matched.

Reference: RFC6895 Domain Name System (DNS) IANA Considerations
To Decept DNS Response: Easy and Simple (2/2)

References:
RFC768 User Datagram Protocol
RFC791 INTERNET PROTOCOL
To Decept TCP Response: Easy and Simple

Reference: RFC793 TRANSMISSION CONTROL PROTOCOL
DQB architecture

1. Client requests IP address of evil.example.com to DNS.
2. DNS responses IP address of evil.example.com to client (192.168.0.1).

- While monitoring request 1, if FQDN in request 1 exists in malicious FQDN list, response 2' is sent to client (172.16.0.1 is included in response 2').

Client receives response 2', and client accesses to 172.16.0.1(3').

...of course, response 2' is ignored.
**Shutdowner Architecture**

1. Client sends SYN packet to Proxy load balancer
2. Proxy load balancer sends SYN+ACK packet to client

While SYN packet monitoring, if SYN packet source IP address is included in Block IP addresses list, Decepted SYN+ACK packet is sent. Of course, sequence number in is not related to sequence number in (Real SYN+ACK).

Client receives packet and related ACK packet. Proxy Load Balancer don’t know the sequence number in, and sends RST packet.

When client receives after, client sends RST packet to Proxy Load Balancer because of receiving SYN+ACK.
Performance?

- 0.055 ~ 0.117 (msec): DQB processing time from receiving packet to sending decepted response packet
  - Real DNS server software processes slower than DQB
- 0.019 ~ 0.023 (msec): Shutdowner processing time from receiving SYN packet to sending decepted SYN+ACK packet
  - Real TCP/IP stack processes slower than Shutdowner
Performance in real environment

- Request packet
- After 0.5msec, Response packet (Decepted packet by DQB)
- After 7.2msec, Response packet (True packet by BIND)

Win!
DQB and Shutdowner hardware spec?

• DQB works on:
  • PowerEdge R230(<$2k)
    • Xeon(R) CPU E3-1271 v3 @ 3.60GHz * 1
    • 16GB of memory
    • Intel I350 GbE NIC(4 ports)

• Shutdowner works on:
  • PowerEdge R230(<$2k)
    • Xeon(R) CPU E3-1271 v3 @ 3.60GHz * 1
    • 16GB of memory
    • Intel I350 GbE NIC(4 ports)
Simple Operation

Complex Systems Operations make works slower
Web application for DQB/Shutdowner Operation

- We developed a Web application for managing DQB and Shutdowner
  - Add / View / Delete malicious FQDN or IP address simply
  - Malicious FQDN/IP address can be added by Web API in Web Application.
Released in 2015

• Almost no trouble for 4 years
After DQB and Shutdowner

Malicious FQDN/IP address of Infected host

Detection of Malicious Host

Our SIEM

Malicious host detection by various of logs

NTTDATA-CERT

IR/Forensics

Log

URL Filter (Proxy) /FireWall/DNS etc

Attacker

Malicious Host

Web

Web

Web

NTTDATA-CERT
After DQB and Shutdowner: Responding to malicious URL got faster

• Find suspicious URL by our SIEM or others
  • Consider whether URL is malicious or not(<1h)
  • Set malicious FQDN got from malicious URL to DQB(<1min)
• Send request URL filter operators set of malicious FQDN once a week
• After URL filter operators work, unset malicious FQDN from DQB
  • Done

• Even if malicious URLs are found oftenly, URL Filter operators don’t receive requests oftenly( once a week )
  • Set
Good point and Better point

• Good point (we thought before deployment)
  • Reducing access from our company to malicious hosts by using DQB
  • Reducing operations by simple Web UI
  • Reducing operations of URL filter operators

• Better point (we didn’t think before deployment)
  • No complaint from users
  • Presentation in Annual FIRST Conference
  • Malware infection detection (partly)

Our contact is shown in sorry page
Limitations and our environment’s case

- Limitations
  - DQB cannot process **DNS request via TCP and DNS over TLS**
  - Decepted response by DQB is ignored **when DNSSEC is used**
  - **When IP address is included** in malicious URL(e.g. http://10.0.0.1/...), DQB don’t work

- In our environment
  - **UDP is used for DNS request, and DNSSEC is disabled**, then DQB works well
  - **Not so much URL including malicious IP address(es) found**

→ No Problem!
Jackpot!

Decepting DNS response makes malware infected hosts accessing to “our” landing host
Example: Responding to Malware infection

• NTT DATA infected Ransomware in 5\textsuperscript{th}, Jan. 2018...
  • Many nodes were crushed by Ransomware infection
• We use large amount of computers
  • above 20k～30k, Windows run most of computers
  • Many of them: \textit{Windows Embedded(without MS17-010 patches)}
• Most of them goes to bluescreened
In NTT DATA’s web site:

Increased access to DQB host

• When Ransomware runs on infected node
  • Attempts “Killswitch” FQDN resolution (by DNS)
  • DQB knows “Killswitch” FQDN and send decepted DNS response
  • DQB works hard, and access to host announced by DQB (we call “Landing Host”) increases
Why too many accesses were made by Ransomware?

• After Infection, access to kill switch occurs
• Multiple Infections, **Multiple Accesses**
DQB work when ransomware is infected

www.iuqerfsodp9ifjaosdfjhgosurijfaewrwegwea.com
a.b.c.d

DNS

www.iuqerfsodp9ifjaosdfjhgosurijfaewrwegwea.com
evil.example.com
IN A a.b.c.d

DQB

10.x.x.x

1

www.iuqerfsodp9ifjaosdfjhgosurijfaewrwegwea.com
IN A 10.x.x.x

Name resolution for killswitch access:
DNS request to resolve IP address of
www.iuqerfsodp9ifjaosdfjhgosurijfaewrwegwea.com

Landing Host(Safe Host)

killswitch access log
Is stored 😊

client

killswitch access

②'

①

③'
Amount of Killswitch Access:

- About 500k accesses
- 6 – 7 accesses to “landing host”
Responding to Ransomware infection

• Isolate network segment that includes infected hosts
• Harden terminal(s) to prevent infection
• Few terminals cannot be responded
  → use DQB log to exploit by using MS17-010(and place the target to bluescreened state)

  10.x.x.x - - [18/Jan/2018:02:31:39 +0900] "GET / HTTP/1.1" 200 670 "-" "-

  www.iuquerfsodp9ifjaaposdfjhgosurijfaewrwegwea.com

  • We developed the auto response tool in a few hours and got effective operation tool
    (and **we got rest time**) 
    • If we have PoC code for exploitation, crushing system is easier than getting shell access
Conclusion

• We don’t need so expensive solution(s), but we need(and developed) our requested tool

• Solution like “DQB” and “Shutdowner” can be developed if you understand network protocol and architecture basis and you can develop some tools by using C and Python 😊

• We may be able to develop tools not only detection, but also response by ourselves😊

• if you know your enemy and know yourself, in a hundred battles you will never be defeated; know your enemy ( from “The Art of War” by Sun Tzu )
Any Question?

Kunio.Miyamoto@nttdata.com
References

This section is not talked but useful to understand DQB and Shutdowner architecture
For example: FQDN matching and frame building strategy

• Matching www.example.com
  www.example.com is described “¥3www¥7example¥3com¥0” in DNS request packet
  →DQB use FQDN “¥3www¥7example¥3com¥0”, and don’t parse “¥3www¥7example¥3com¥0” to “www.example.com”.

• Prebuilt DNS response except IP addresses, Port numbers and ID
  Don’t build packet fully and dynamically
Design for Performance

• Programming Language: C

• On-memory processing and intend CPU cache
  main loop: smaller, no library call (of course, systemcalls are not library call)

• Logging to shared memory(and write file by logger process)

• Don’t use malloc() timely, use malloc() for entire use at first.
  To prevent bugs caused by memory management mistaken

• Lock free(for delay prevention caused by scheduler)
  use flags instead of (any kind of) lock

• Read packet header +α only
Design for Performance

• Don’t use async processing for socket() to assure the time syscall finished is the time decepted packet is sent
  ```c
  int asyn_flag = 0;
  // (snip)
  fd = socket(PF_PACKET, SOCK_RAW, htons(ETH_P_ALL));
  (void)ioctl(fd, FIOASYNC, &asyn_flag);
  ```

• If you can, DPDK is suitable for DQB and Shutdowner

• Shutdowner: use bloom filter, use pointer array(use syntax like “if” or switch ... case in IP address matching)
  ```c
  void *func[IP address space];
  func[] is valiable that contains function to send decepted syn+ack packet or do nothing when packet is received, call func[srcIP]
  ```

• DQB: Don’t process dynamically like FQDN parse, and preprocess to build response packet framework
Traffic Generator by using Linux pktgen

#!/bin/sh

modprobe pktgen

echo "rem_device_all" > /proc/net/pktgen/kpktgend_0
echo "add_device eth2" > /proc/net/pktgen/kpktgend_0
echo "count 100000" > proc/net/pktgen/eth2
echo "clone_skb 1" > /proc/net/pktgen/eth2
echo "pkt_size 60" > /proc/net/pktgen/eth2
# delay specified value (nanosec order)
echo "delay 20000" > /proc/net/pktgen/eth2
echo "src_min 172.16.0.2" > /proc/net/pktgen/eth2
echo "src_max 172.16.0.2" > /proc/net/pktgen/eth2
echo "src_mac a0:36:9f:a8:8a:3c" > /proc/net/pktgen/eth2
echo "dst 172.16.0.3" > /proc/net/pktgen/eth2
echo "dst_mac a0:36:9f:a8:86:b8" > /proc/net/pktgen/eth2
echo "start" > /proc/net/pktgen/pgctrl

cat /proc/net/pktgen/eth2