

SGS-Thomson Microelectronics

STMicroelectronics

- Partita IVA: 00951900968 - Codice Fiscale: 09291380153
- Rag. Sociale: STMICROELECTRONICS S.R.L.
- Indirizzo: VIA CAMILLO OLIVETTI 2 - 20864 - AGRATE BRIANZA (MB)
- Rea: 1281765
- PEC: st_srl@legalmail.it
- Fatturato: € 1.939.529.000,00 (2021)
- Dipendenti : 12166 (2023)

Secondo le nostre ricerche ST Microelectronics appartiene a BNP Paribas Asset Management Holding S.A

STM conta su un organico composto da più di **50.000 risorse**, impiegate nelle numerose sedi e fabbriche aziendali distribuite a livello mondiale. **In Italia, STMicroelectronics è presente con 11 siti**, tra cui gli stabilimenti produttivi di:

Agrate Brianza

Catania

Marcianise

Durante le nostre ricerche scopriremo le numerose sedi in Francia e nel resto dell'europa.

Siti E-Commerce del mondo STM - **Distributori Globali**

Elettronica Digikey: www.digikey.com

Farnell / Newark / element14, una società Avnet: www.farnell.com

Mouser Electronics: www.mouser.com

Riguardo i siti e-commerce di distribuzione globale che utilizza ST ho deciso di non approfondire più di tanto.

Sono comunque probabilmente presenti vulnerabilità su più di un sito.

Senza entrare troppo nello specifico ci soffermiamo al primo sito:

www.digikey.com

digikey.com

Technology stack

Static site generator



Next.js (11.0.1)

Next.js 11.0.1 presenta le seguenti vulnerabilità:

Esaurimento delle risorse

User Interface (UI) Misrepresentation of Critical information

Scripting tra siti (XSS)

Open Redirect



DataTables (1.10.12)

Data Tables 1.10.12 è una versione obsoleta, presenta le seguenti vulnerabilità:

Scripting tra siti (XSS)

Prototype Pollution

Eseguiamo una breve ricerca di numeri telefonici associati a STMicroelectronics

phonebook:Stmicroelectronics numrange:3000000000-600000000000

Unified Patents
https://portal.unifiedpatents.com · Traduci questa pagina

US-20080070549-A1 - Method for Setting a Key and a ...
US-20050153740-A1 · Priority Date: 2004-01-13 · Assignees: Motorola Solutions Inc · Title: Linked Storage for Enhanced Phone Book Entries in Mobile Communications ...

italchamber.org.sg
https://www.italchamber.org.sg/files/content-files/PDF

ICCS DIRECTORY 21/22
STMICROELECTRONICS Asia Pacific Pte Ltd ... T (86) 13818736209 (Shanghai). E lr@rsatx.com. M (65) 9650 4445. E andrea.resmini@hotmail.it. Page 211. 418. 419.
219 pagine

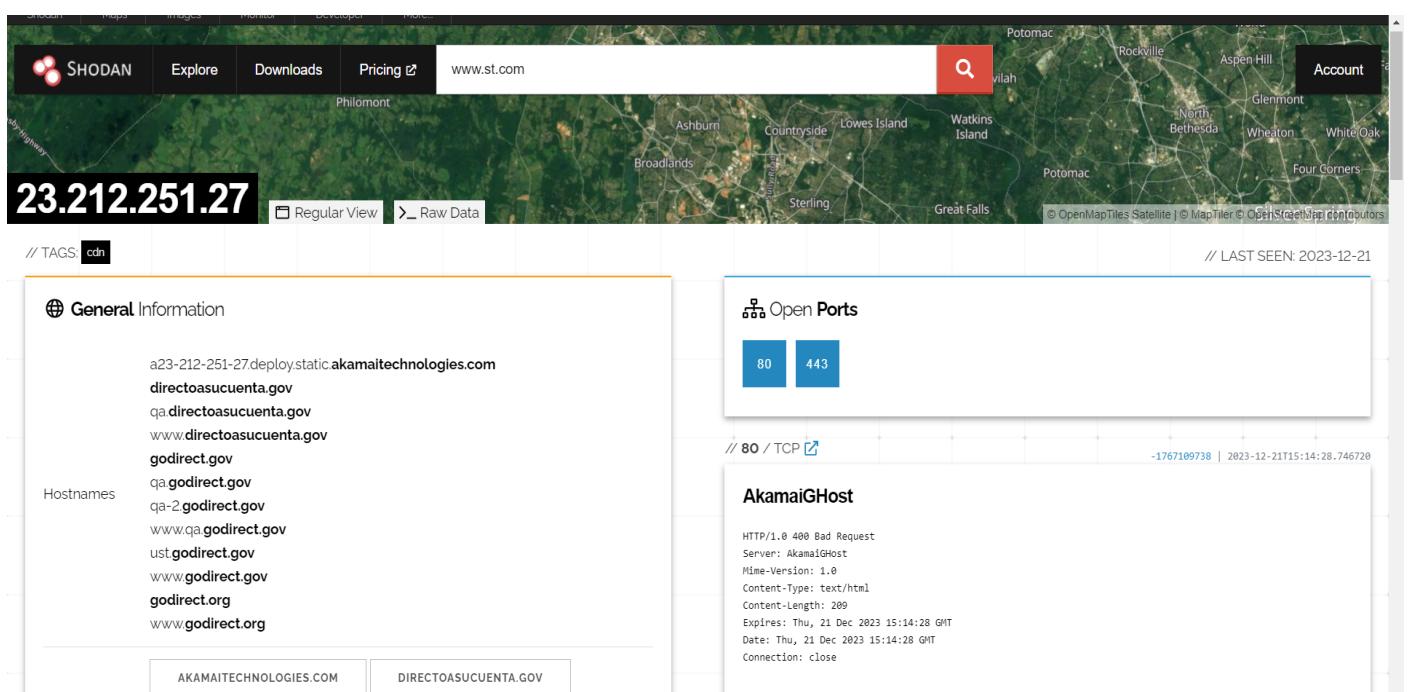
United States International Trade Commission (.gov)
https://www.usitc.gov/publications/pub4386/PDF

Certain Semiconductor Chips and Products Containing Same
25 lug 2012 — STMicroelectronics NV of Geneva, Switzerland; STMicroelectronics Inc. ...
233,239,244,249 (nVidia accused products); id at ~257, 259, 280, 286 ...
466 pagine



Tramite queste ricerche siamo riusciti a trovare numeri di telefono, e-mail e dati sensibili riguardanti alcuni dei personaggi più importante di questa multinazionale, quello che però mi è subito chiaro è che i dipendenti seguono un corso di sicurezza informatica ogni tot di tempo, la loro formazione non ci permette di ottenere dati sensibili riguardanti l'azienda e le loro connessioni interne

Fra le informazioni che abbiamo trovato durante una prima ricerca sappiamo che il sito di ST Microelectronics è: www.st.com
Facciamo una breve ricerca su Shodan:



SHODAN Explore Downloads Pricing ↗ www.st.com Search Account

23.212.251.27 Regular View Raw Data

// TAGS: cdn

// LAST SEEN: 2023-12-21

General Information

Hostnames

a23-212-251-27.deploy.static.akamaitechnologies.com
directoasuenta.gov
qa.directoasuenta.gov
www.directoasuenta.gov
godirect.gov
qa.godirect.gov
qa-2.godirect.gov
www.qa.godirect.gov
ust.godirect.gov
www.godirect.gov
godirect.org
www.godirect.org

AKAMAITECHNOLOGIES.COM DIRECTOASUENTA.GOV

Open Ports

80 443

80 / TCP

AkamaiGHost

HTTP/1.0 400 Bad Request
Server: AkamaiGHost
Mime-Version: 1.0
Content-Type: text/html
Content-Length: 209
Expires: Thu, 21 Dec 2023 15:14:28 GMT
Date: Thu, 21 Dec 2023 15:14:28 GMT
Connection: close

Facciamo una ricerca più approfondita utilizzando Wappalyzer:
Questo tool ci segnali i possibili software e le relative versioni installate sulla macchina che si osserva.

Questo è un piccolo quadro delle criticità rilevate:

Load balancers

-  Amazon ELB

Amazon Elb è un Application Load Balancer, questo rende impossibile attaccare tramite attacco DDoS sperando di buttare giù il server in quanto; Quando un server sarà totalmente pieno un altro server si aprirà per ospitare nuovi utenti senza preavviso.

JavaScript libraries

 Moment.js (2.18.1)	 jQuery (2.1.1)
 Modernizr (2.8.3)	 Browser-Update.org (3.3.49)
 core-js (3.27.0)	 LazySizes
 Lodash (4.17.4)	 jQuery UI (1.12.1)
 DataTables (1.10.15)	 Boomerang
 web-vitals	 List.js

Alcune di queste versioni sembrano essere obsolete, dunque eseguiamo delle brevi ricerche che ci portano a pensare che:

Un criminale informatico, Black Hat potrebbe facilmente violare i sistemi.

Un metodo di attacco, che si ripete di continuo durante le ricerche di eventuali vulnerabilità, risulta essere XSS.

Sembra sia utilizzato **jQuery 2.1.1**, è una versione obsoleta e vulnerabile ai seguenti attacchi:

Cross-site Scripting (XSS)

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting tra siti (XSS).

Passare HTML da fonti non attendibili - anche dopo averlo disinfeccato - a uno dei metodi di manipolazione DOM di jQuery (ad es. `.html()`, `.append()`, e altri) possono eseguire codice non attendibile.

Un altro possibile attacco XSS da parte di un malintenzionato potrebbe essere:

Passaggio di HTML contenente `<option>` elementi da fonti non attendibili - anche dopo averlo disinfeccato - a uno dei metodi di manipolazione DOM di jQuery (ad es. `.html()`, `.append()`, e altri) possono eseguire codice non attendibile.

A questa vulnerabilità è stato assegnato **CVE-2020-23064**

Vulnerabilità Angular 1.3.5

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting tra siti (XSS) tramite richieste **JSONP**, a causa di una sanificazione impropria degli URL delle risorse

Le versioni interessate di questo pacchetto sono vulnerabili alla negazione del servizio di espressioni regolari (ReDoS) tramite `angular.copy()` funzione di utilità dovuta all'uso di un'espressione regolare non sicura. Lo sfruttamento

di questa vulnerabilità è possibile attraverso un ampio input accuratamente realizzato, che può provocare un backtracking catastrofico.

Le versioni interessate di questo pacchetto sono vulnerabili alla negazione del servizio di espressioni regolari (**ReDoS**) tramite `$resource` servizio dovuto all'uso di un'espressione regolare insicura. Lo sfruttamento di questa vulnerabilità è possibile attraverso un ampio input accuratamente realizzato, che può provocare un backtracking catastrofico.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting transitivo (**XSS**) a causa della memorizzazione nella cache di pagine non sicura nel browser Internet Explorer, che consente l'interpolazione di `<textarea>` elementi.

XSS può anche essere attivato in applicazioni AngularJS che disinettano i frammenti HTML controllati dall'utente prima di passarli a `JQLite` metodi come `JQLite.prepend`, `JQLite.after`, `JQLite.append`, `JQLite.replaceWith`, `JQLite.append`, `new JQLite` e `angular.element`.

La libreria di manipolazione `JQLite` (DOM che fa parte di AngularJS) manipola l'HTML di input prima di inserirlo nel DOM in `jqLiteBuildFragment`.

Una delle modifiche eseguite espande un tag di auto-chiusura XHTML.

Se `jqLiteBuildFragment` si chiama (ad es. `via new JQLite(aString)`) con stringa HTML controllata dall'utente che è stata disinettata (ad es. con `DOMPurify`), la trasformazione effettuata da `JQLite` può modificare alcune forme di payload inerte e sanificato in un payload contenente JavaScript e attivare un XSS quando il payload viene inserito in DOM.

PoC

```
const inertPayload = `<div><style><style/><img src=x
onerror="alert(1337)"/>`
```

Si noti che l'elemento style non è chiuso e <img sarebbe un nodo di testo all'interno dello stile se inserito nel DOM così com'è. Pertanto, alcuni disinfettanti HTML lascerebbero il <img com'è senza elaborarlo e spogliare il onerror attributo.

```
angular.element(document).append(inertPayload);
```

Questo aviserà, come <style/> sarà sostituito con <style></style> prima di aggiungerlo al DOM, chiudendo presto l'elemento style e riattivando img.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting transitivo (XSS). La sostituzione HTML di input basata su regex può trasformare il codice disinfettato in uno non autorizzato. avvolgimento <option> elementi in <select> quelli cambiano il comportamento di analisi, portando a un codice forse non igienizzante.

Le versioni interessate di questo pacchetto sono vulnerabili a Denial of Service (DoS). Nessuna

Altra vulnerabilità, possibile attacco XSS

I browser mutano valori di attributi come 　 javascript:alert(1) quando sono scritti nel DOM via innerHTML in vari modi specifici del fornitore. In Chrome (< 62), questa mutazione ha rimosso il precedente "whitespace" risultando in un valore che potrebbe finire per essere eseguito come JavaScript.

Ecco un esempio di ciò che potrebbe accadere:

```
// Code goes here

var h1 = document.querySelector('h1');

h1.innerHTML = '<a
href="#&#x3000;javascript:alert(1)">CLICKME</a>';

var innerHTML = h1.innerHTML;

console.log(innerHTML);

h1.innerHTML = innerHTML;
```

Il disinfettante contiene un po 'di codice che innesca questa mutazione su un pezzo inerte di DOM, prima che l'angolare lo disinfetta.

Nota: Chrome 62 non sembra più mutare questa particolare stringa, invece lascia semplicemente il "whitespace" in posizione. Ciò significa probabilmente che Chrome 62 non è più vulnerabile a questo vettore di attacco specifico.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting tra siti (**XSS**). Il `$http` il servizio consente richieste JSONP con URL non attendibili, che potrebbero essere sfruttate da un utente malintenzionato.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting tra siti (**XSS**) tramite file SVG se `enableSvg` è impostato.

Le versioni interessate di questo pacchetto sono vulnerabili a **JSONP Callback Attack**. JSONP (JSON con riempimento) è un metodo utilizzato per richiedere dati a un server residente in un dominio diverso rispetto al client.

Qualsiasi URL può eseguire richieste JSONP, consentendo il pieno accesso al browser e al contesto JavaScript. Ciò può portare allo scripting tra siti.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting tra siti (XSS) a causa del `usemap` attributo non inserito nella lista nera.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting tra siti (XSS) tramite SVG `<use>` elemento. Il `<use>` element può fare riferimento alla stessa origine (di SVG esterna) e può includere `xlink:href` URL javascript o oggetti estranei che possono eseguire XSS. La modifica non consente `<use>` elementi nel markup SVG sanificato. Un esempio di un documento SVG dannoso sarebbe:

SVG per disinfettare:

```
<svg><use xlink:href="test.svg#xss" /></svg>
```

File SVG esterno (test.svg):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg xmlns:svg="http://www.w3.org/2000/svg"
      xmlns="http://www.w3.org/2000/svg" width="100"
      height="100"
      id="xss">
<a xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:href="javascript:alert(1)">
  <circle cx="50" cy="50" r="40" stroke="black"
  stroke-width="3" fill="red" />
</a>
</svg>
```

Qui l'SVG per disinfettare i carichi nel `test.svg` file tramite il `<use>` elemento. Il disinfettante non è in grado di analizzare questo file, che contiene un mark-up eseguibile dannoso. Questo può essere sfruttato solo se il file esterno è disponibile tramite stesse restrizioni di origine in atto.

Le versioni interessate di questo pacchetto sono vulnerabili agli attacchi Cross-site Scripting (XSS) che coinvolgono l'assegnazione `constructor` proprietà.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting trasi (**XSS**). Questo errore si verifica quando `$sanitize` disinettante tenta di controllare l'input per un possibile payload mXSS e gli errori di verifica dovuti all'input che muta indefinitamente. Questo potrebbe essere un segno che il payload contiene codice che sfrutta una vulnerabilità mXSS nel browser.

mXSS attack sfrutta i bug del browser che causano l'analisi di alcuni browser in DOM, che una volta serializzato non corrisponde all'input originale. Questi bug del browser possono essere sfruttati dagli aggressori per creare payload che sembra innocuo per i disinettanti, ma a causa delle mutazioni causate dal browser vengono trasformati in codice pericoloso una volta elaborati dopo la sanificazione.

Le versioni interessate di questo pacchetto sono vulnerabili a **Clickjacking**. Abilitando l'impostazione SVG senza prendere altre precauzioni, è possibile esporre l'applicazione agli attacchi di dirottamento dei clic. In questi attacchi, gli elementi SVG sanificati potrebbero essere posizionati al di fuori dell'elemento contenente ed essere resi su altri elementi nella pagina (ad es. un link di accesso). Tale comportamento può quindi provocare incidenti di phishing.

Per proteggere da questi, impostare esplicitamente `overflow: hidden` regola css per tutti i potenziali tag SVG all'interno del contenuto sanificato:

```
.rootOfTheIncludedContent svg {  
    overflow: hidden !important;  
}
```

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting trasi (**XSS**) a causa della corretta sanificazione di `xlink:href` attributi.

Le versioni interessate di questo pacchetto sono vulnerabili all'esecuzione del codice arbitrario tramite tag di animazione svg non sicuri.

jQuery 2.1.1

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting transitivo (XSS). Passare HTML da fonti non attendibili - anche dopo averlo disinfeccato - a uno dei metodi di manipolazione DOM di jQuery (ad es. `.html()`, `.append()`, e altri) possono eseguire codice non attendibile.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting transitivo (XSS) Passaggio di HTML contenente `<option>` elementi da fonti non attendibili - anche dopo averlo disinfeccato - a uno dei metodi di manipolazione DOM di jQuery (ad es. `.html()`, `.append()`, e altri) possono eseguire codice non attendibile.

Le versioni interessate di questo pacchetto sono vulnerabili a Prototype Pollution. Il `extend` la funzione può essere indotta nel modificare il prototipo di `Object` quando l'attaccante controlla parte della struttura passata a questa funzione. Ciò può consentire a un utente malintenzionato di aggiungere o modificare una proprietà esistente che quindi esisterà su tutti gli oggetti.

Nota: CVE-2019-5428 è un duplicato di CVE-2019-11358

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting transitivo (XSS) attacchi quando viene eseguita una richiesta ajax tra domini senza `dataType` opzione che causa `text/javascript` risposte da eseguire.

Nota: Dopo essere stata implementata nella versione 1.12.0, la correzione di questa vulnerabilità è stata ripristinata in 1.12.3, quindi è stata reintrodotta solo nella versione 3.0.0-beta1. La correzione non è mai stata rilasciata in nessun tag del ramo 2.x.x, poiché è stata ripristinata dal ramo prima di essere rilasciata.

Nota: CVE-2017-16012 è un duplicato di CVE-2015-9251

Vulnerabilità di moment.js 2.18.1

Le versioni interessate di questo pacchetto sono vulnerabili alla negazione del servizio di espressioni regolari (ReDoS) tramite `preprocessRFC2822()` funzione in `from-string.js`, durante l'elaborazione di una stringa elaborata molto a lungo (oltre 10k caratteri).

Le versioni interessate di questo pacchetto sono vulnerabili a Directory Traversal quando un utente fornisce una stringa locale che viene utilizzata direttamente per cambiare la locale del momento.

Le versioni interessate di questo pacchetto sono vulnerabili a Regular Expression Denial of Service (ReDoS). Ha usato un'espressione regolare `(/[0-9]*['a-z\u00A0-\u05FF\u0700-\uD7FF\uF900-\uFDCE\uFDF0-\uFFEF]+|[\u0600-\u06FF\//]+(\s*?[\u0600-\u06FF]+){1,2})/i` per analizzare le date specificate come stringhe. Ciò può causare un impatto molto basso di circa 2 secondi di tempo di corrispondenza per dati lunghi 50k caratteri.

Vulnerabilità di **lodash 4.17.4**

Iniezione di comando

Hodash è una moderna libreria di utilità JavaScript che offre modularità, prestazioni ed extra.

Le versioni interessate di questo pacchetto sono vulnerabili a Command Injection via `template`.

PoC

```
var _ = require('lodash');

_.template('', { variable: '' }){console.log(process.env)};
with(obj' })()
```

Le versioni interessate di questo pacchetto sono vulnerabili alla negazione del servizio di espressioni regolari (ReDoS) tramite `toNumber`, `trim` e `trimEnd` funzioni.

POC

```
var lo = require('lodash');

function build_blank (n) {
var ret = "1"
for (var i = 0; i < n; i++) {
ret += " "
}
```

```
return ret + "1";
}

var s = build_blank(50000)
var time0 = Date.now();
lo.trim(s)
var time_cost0 = Date.now() - time0;
console.log("time_cost0: " + time_cost0)

var time1 = Date.now();
lo.toNumber(s)
var time_cost1 = Date.now() - time1;
console.log("time_cost1: " + time_cost1)

var time2 = Date.now();
lo.trimEnd(s)
var time_cost2 = Date.now() - time2;
console.log("time_cost2: " + time_cost2)
```

Le versioni interessate di questo pacchetto sono vulnerabili all'inquinamento da prototipo tramite `setWith` e `set` funzioni.

Le versioni interessate di questo pacchetto sono vulnerabili a Prototype Pollution. La funzione `zipObjectDeep` può essere indotto ad aggiungere o modificare le proprietà del prototipo Object. Queste proprietà saranno presenti su tutti gli oggetti.

Le versioni interessate di questo pacchetto sono vulnerabili a Prototype Pollution. La funzione `defaultsDeep` potrebbe essere indotto ad aggiungere o modificare proprietà di `Object.prototype` usando a constructor payload.

Le versioni interessate di questo pacchetto sono vulnerabili a Regular Expression Denial of Service (ReDoS). Analizza le date usando le stringhe regex, che possono causare un rallentamento di 2 secondi per 50k caratteri.

Le versioni interessate di questo pacchetto sono vulnerabili a Prototype Pollution. Le funzioni `merge`, `mergeWith`, e `defaultsDeep` potrebbe essere

indotto ad aggiungere o modificare proprietà di `Object.prototype`. Ciò è dovuto a una correzione incompleta a CVE-2018-3721.

Le versioni interessate di questo pacchetto sono vulnerabili a Prototype Pollution. La funzione utility consente la modifica del `Object` prototipo. Se un utente malintenzionato può controllare parte della struttura passata a questa funzione, potrebbe aggiungere o modificare una proprietà esistente.

Vulnerabilità JQuery-UI 1.12.1

`jquery-ui` è una libreria per la manipolazione di elementi dell'interfaccia utente tramite `jQuery`.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting trascritti (**XSS**) tramite l'inizializzazione di `checkboxradio` widget su un tag di input racchiuso in un'etichetta, che porta al contenuto dell'etichetta principale considerato come l'etichetta di input.

Sfruttare questa vulnerabilità è possibile se a `.checkboxradio("refresh")` la chiamata viene eseguita su tale widget e l'HTML iniziale contiene entità HTML codificate, portandole a essere erroneamente decodificate.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting trascritti (**XSS**) quando si accetta il valore del `of` opzione del `.position()` utilizzando da fonti non attendibili che possono portare all'esecuzione di codice non attendibile.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting trascritti (**XSS**) quando si accetta il valore di `altField` opzione del Datepicker widget da fonti non attendibili, che può portare all'esecuzione di codice non attendibile.

POC

Inizializzazione del "datepicker" nel modo seguente:

```
$( "#datepicker" ).datepicker( {
```

```
        altField: "<img onerror='doEvilThing()' src='/404'>",
    } );
}
```

chiamerà il `doEvilThing` funzione.

Le versioni interessate di questo pacchetto sono vulnerabili allo scripting trassiti (XSS). Quando si accetta il valore di vari `*Text` opzioni del Datepicker widget da fonti non attendibili può portare all'esecuzione di codice non attendibile.

###POC Inizializzazione di "Datepicker" nel modo seguente:

```
$( "#datepicker" ).datepicker( {
    showButtonPanel: true,
    showOn: "both",
    closeText: "<script>doEvilThing( 'closeText XSS'
)</script>",
    currentText: "<script>doEvilThing( 'currentText XSS'
)</script>",
    prevText: "<script>doEvilThing( 'prevText XSS'
)</script>",
    nextText: "<script>doEvilThing( 'nextText XSS'
)</script>",
    buttonText: "<script>doEvilThing( 'buttonText XSS'
)</script>",
    appendText: "<script>doEvilThing( 'appendText XSS'
)</script>",
} );
}
```

chiamerà il `doEvilThing()` funzione.

Analizziamo la rete del sito tramite Maltego:



Sarà importante ottenere informazioni di valore, cerchiamo dunque i file destinati ad un tipo specifico di utente.

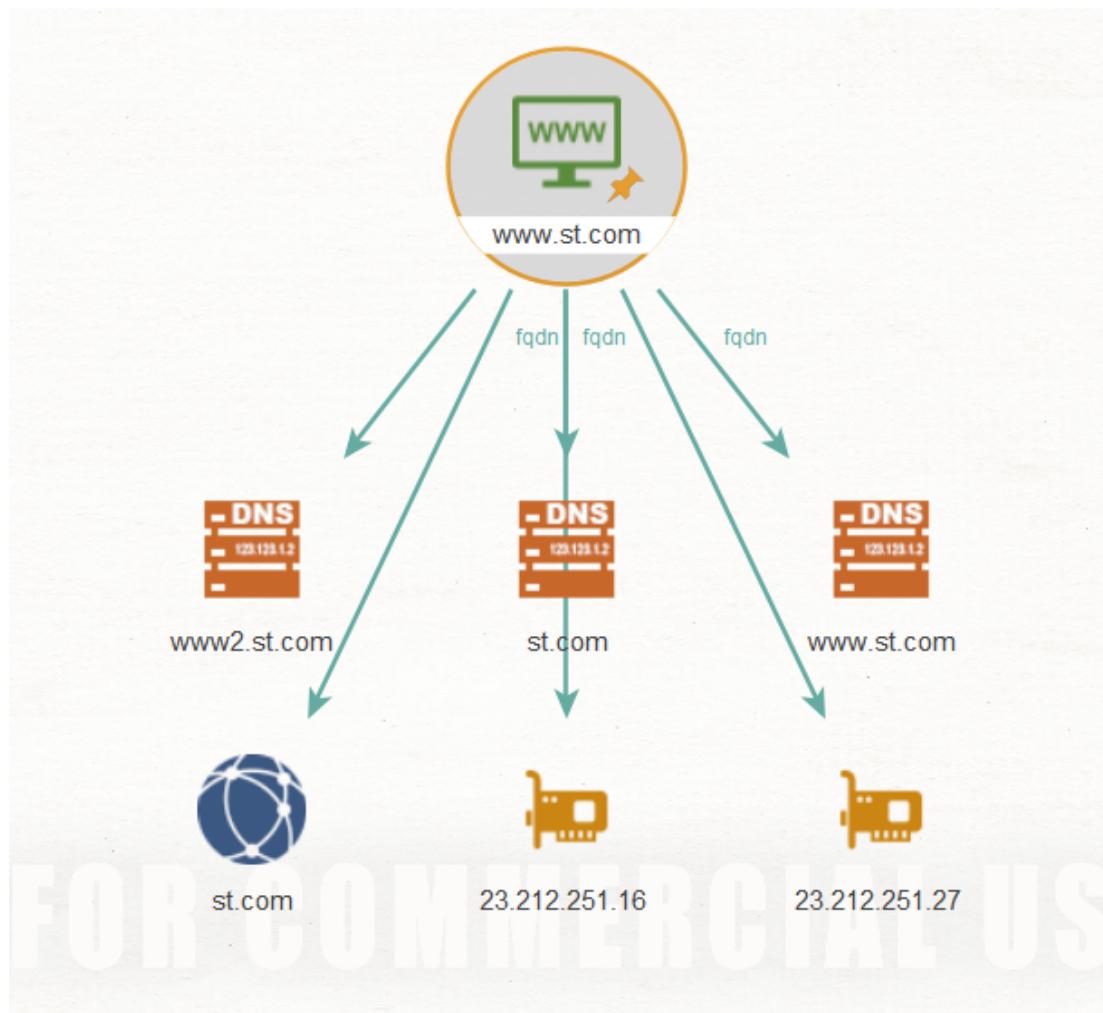
Required inputs X

The following transforms require inputs:

- To Snapshots of Files (Extensions) [Wayback Machine]
 - * Search Extensions (maximum of 20)
 - Search Extensions (maximum of 20)
- To Snapshots of Files (MimeType) [Wayback Machine]
 - * Search Mimetype
 - Search Mimetype
- To Snapshots Containing Phrase [Wayback Machine]
 - * Search Keyword
 - Search Keyword
 - * HTTP status code when the snapshot was taken (set 0 for any status code)
 - HTTP status code when the snapshot was taken (set 0 for any status code)
- To Snapshots between Dates [Wayback Machine]
 - * Search Date Range
 - Search Date Range
- To DNS Name [Enumerate hostname numerically]
 - Dodging
 - Remember these settings

Run! Cancel

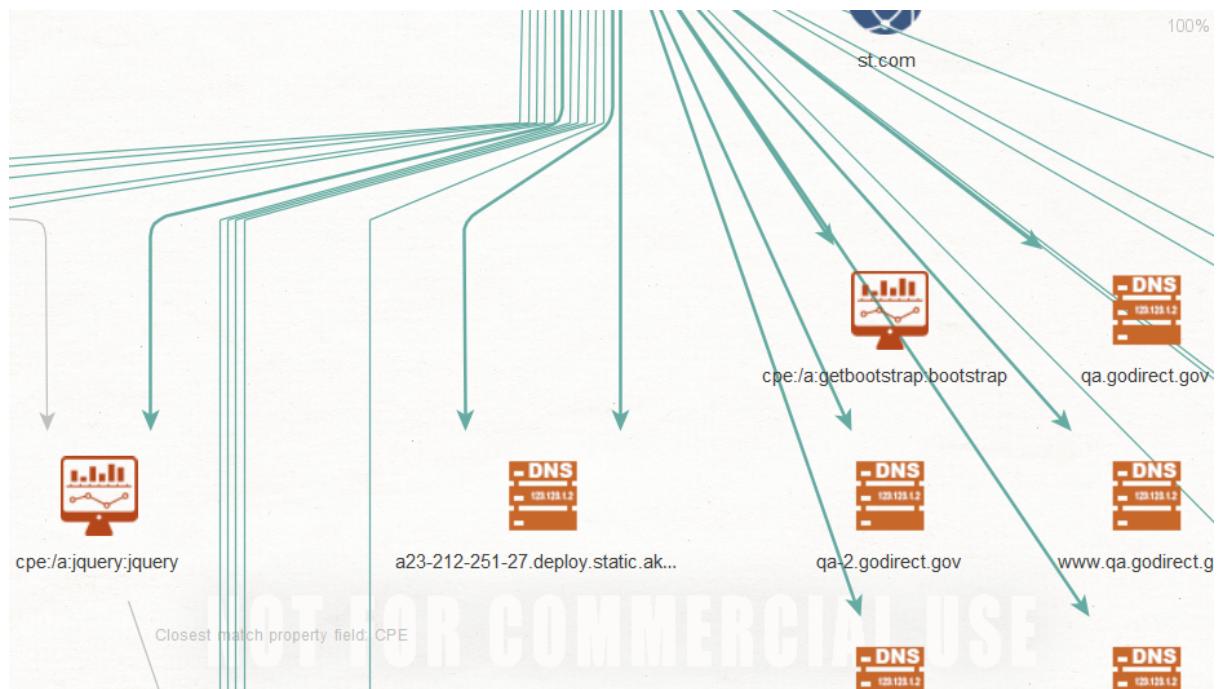
Questi sono i risultati che otteniamo:



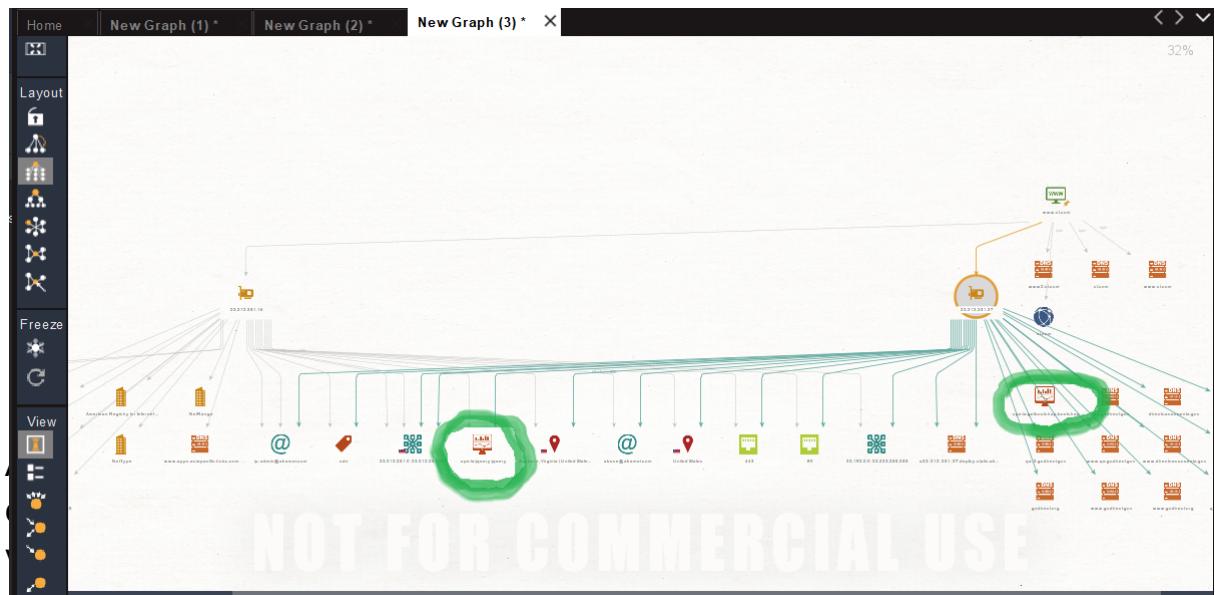
23.212.251.27 Si rivelerà essere l'ip dei Server JQueri e Bootstrap

Su questa scheda abbiamo trovato i server vulnerabili che cercavamo, cioè i server di jQuery e di Bootstrap.

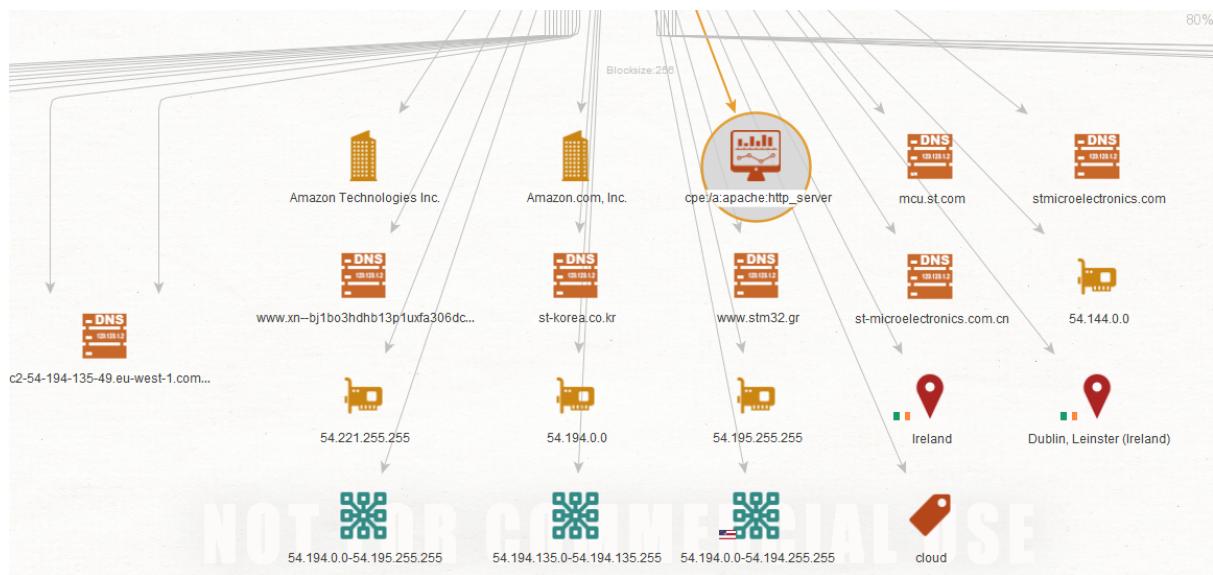
Essi sono i server che un malintenzionato utilizzerebbe per attaccare la rete STm, come evidenziato in precedenza infatti questi server girano probabilmente con software obsoleti, di conseguenza sono possibili attacchi informatici mirati.



Diamo un'occhiata allo schema generale della rete, sulla sinistra il server JQuery, sulla destra il server BootStrap entrambi collegati alle porte 80 e 443



Con molta dedizione anche trovare il server Apache non è un problema, riusciamo dunque ad averne anche gli IP

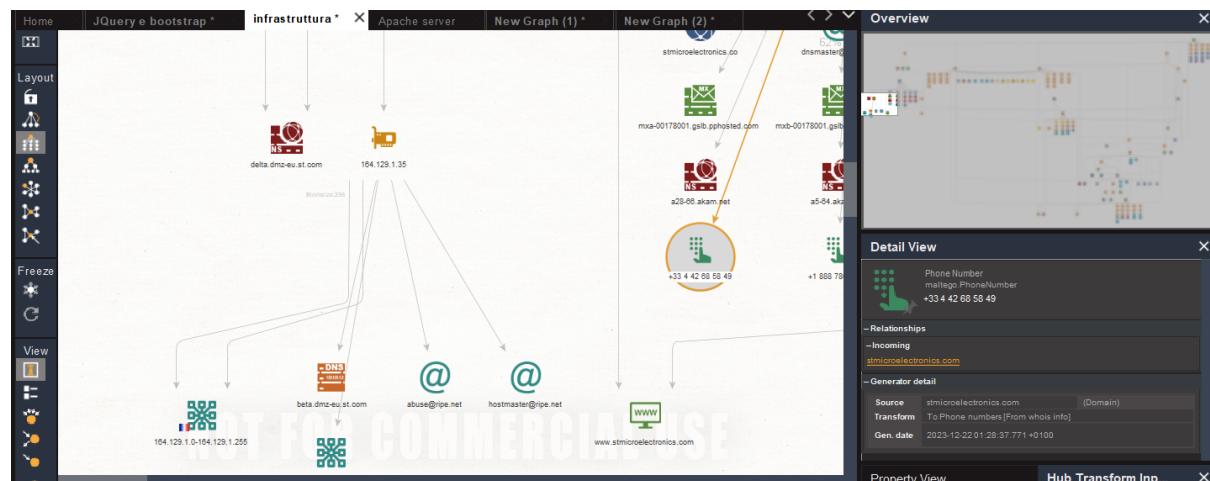


Dopo una lunga ricerca abbiamo però rilevato una rete DMZ che ha accesso al dominio `stmicroelectronics.com`

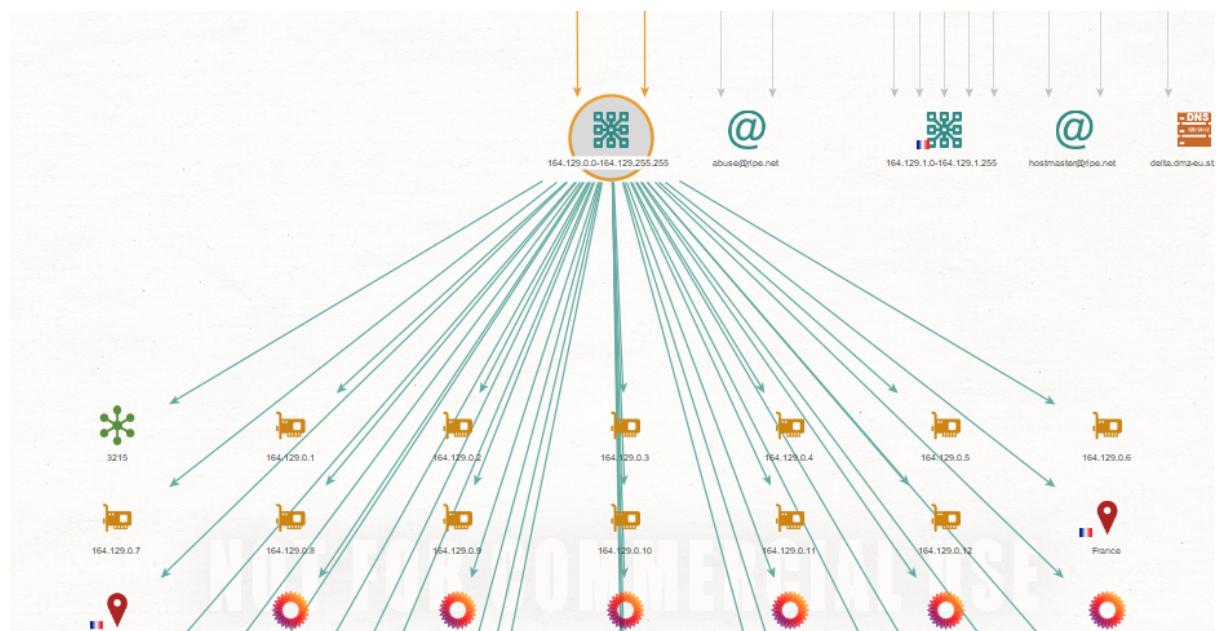
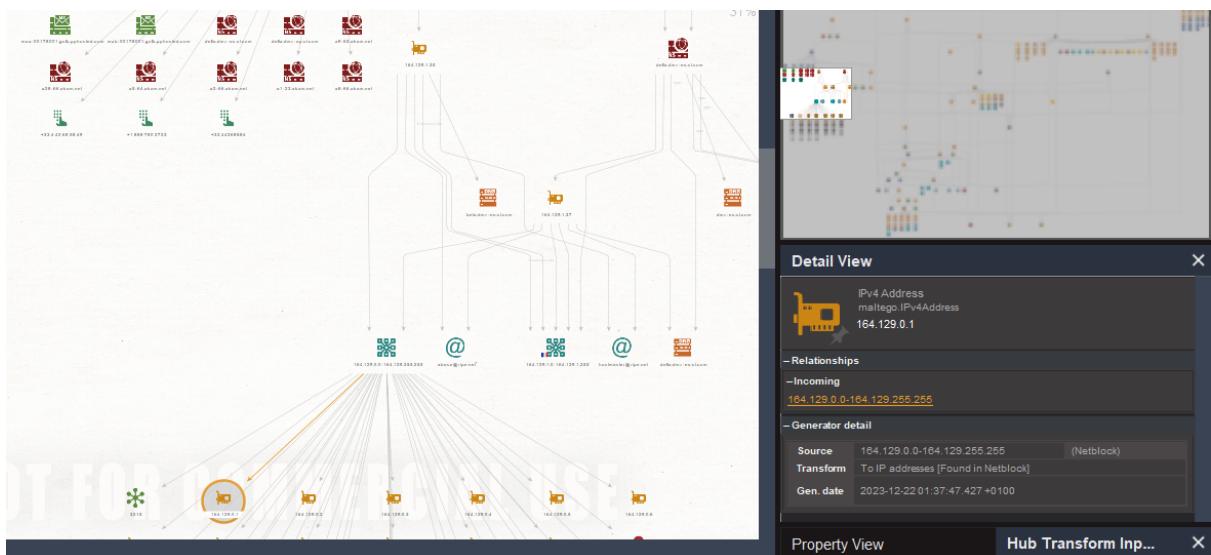
la sua posizione geografica è in Francia, pensiamo ce ne siano molte altre nella rete St e che in seguito ad una ricerca approfondita sarebbe facile trovare una particolare rete DMZ che si vuole attaccare.

Come già spesso sottolineato, i software utilizzati sono obsoleti e tramite una vulnerabilità fra le tante sarebbe possibile utilizzare il Traversal directory per prendere il controllo del sistema informatico.

Giunti a questa conclusione possiamo ipotizzare che anche la DMZ potrebbe accusare attacchi alla sua rete, in quanto esposta anche se protetta



Le informazioni sulla DMZ ci portano a risalire ad indirizzi IP appartenenti alla rete DMZ.

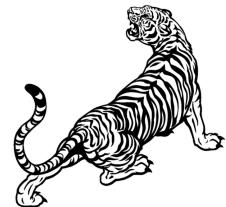


CONCLUSIONI

L'Osint su ST Microelectronics ci porta a supporre sia necessario prendere provvedimenti riguardanti le versioni obsolete non aggiornate dei software utilizzati, affinché un Criminale Informatico, Black hat, non possa utilizzare le sopracitate vulnerabilità con lo scopo di danneggiare economicamente l'infrastruttura di ST.

La nostra indagine ha evidenziato che oltre le vulnerabilità sul sito www.st.com non sono presenti particolari vulnerabilità.

In una relazione fra la fatica di risolvere le problematiche e i danni che un attacco informatico può apportare all'azienda, si consiglia un uso immediato degli aggiornamenti disponibili



White Tiger Do