

Ataques Avançados contra CPL (Control Panel Applets)

Dados da publicação:

Autor: Mauro Risonho de Paula Assumpção

Título: Série: Ataques Avançados 01 de 21 - CPL (Control Panel Applet) Attacks

Mês de Referência: Agosto

Ano de Referência: 2015

Área: Projetos - Segurança

Categoria: Segurança

TAGS (Marque todas as apropriadas):

<input type="checkbox"/> CommVault	<input type="checkbox"/> NetApp	<input type="checkbox"/> Firewalls de Próxima Geração	<input type="checkbox"/> Armaz. Inteligente
<input type="checkbox"/> Citrix	<input type="checkbox"/> Palo Alto	<input checked="" type="checkbox"/> Segurança	<input checked="" type="checkbox"/> Virtualização
<input type="checkbox"/> F5	<input type="checkbox"/> Riverbed	<input type="checkbox"/> Entrega de Aplicações	<input type="checkbox"/> Backup Inteligente
<input type="checkbox"/> ForeScout	<input type="checkbox"/> VMware	<input type="checkbox"/> DDI - DNS	<input type="checkbox"/> IP Address Management
<input type="checkbox"/> Infoblox	<input checked="" type="checkbox"/> Microsof	<input type="checkbox"/> Cloud Computing	<input type="checkbox"/> DHCP
	<input type="checkbox"/> Nutanix		

Palavras-chave:

Windows, Control Panel Applets, CPL, Attacks CPL, Ataques em arquivos CPL, APT, APT (Advanced Persistent Threat), Ameaças Digitais, Ameaças, Ataques Avançados, Segurança, Security, Endpoint, Palo Alto Networks Traps, Traps, Painele de Controle, Windows, Insegurança.

Olho:

Artigo explicativo que informa sobre ataques com CPL (Control Panel Applets) Attacks. O que são arquivos CPL - <https://support.microsoft.com/pt-br/kb/192806>



Artigo:

1.1 O que são e como funcionam CPL (Control Panel Applet):

No sistema operacional Windows, a extensão de arquivo **CPL** é atribuída aos Control Panel Applets, que são miniaplicativos utilizados pelo Painel de Controle introduzidos no Windows 3.x, mas ainda utilizados em versões atuais como o Windows 10.

O Painel de Controle carrega estes miniaplicativos e exibe ícones para que estes sejam acessados facilmente.

Exemplos de Painéis de Controle de várias versões de Windows:

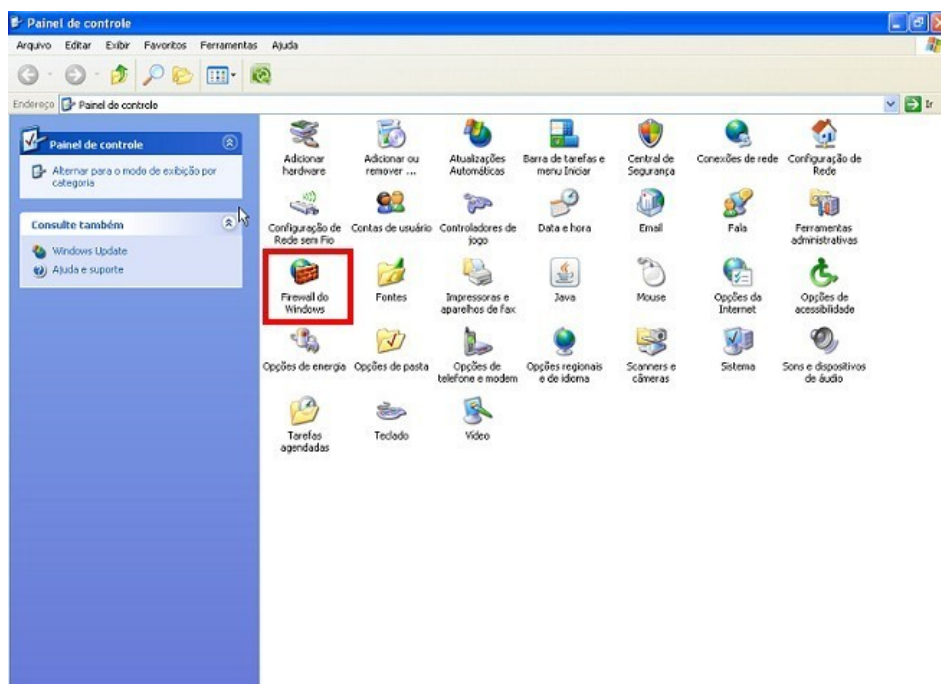


Figura 1: Painel de Controle do Microsoft Windows XP e respectivos arquivos CPL (Control Panel Applet)



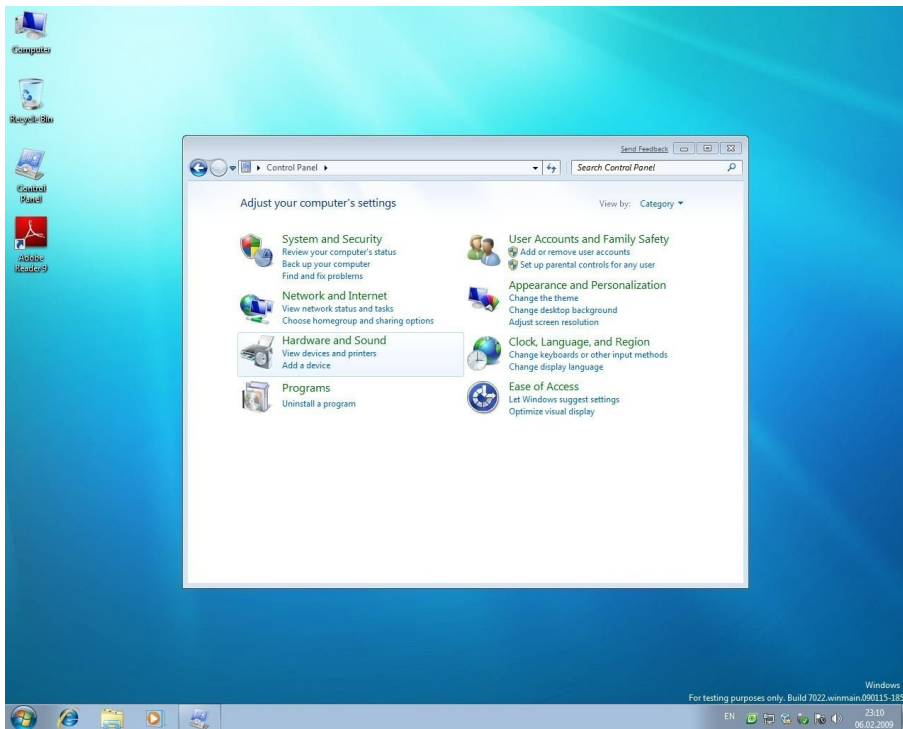


Figura 2: Painel de Controle do Microsoft Windows 7 e respectivos arquivos CPL (Control Panel Applet)

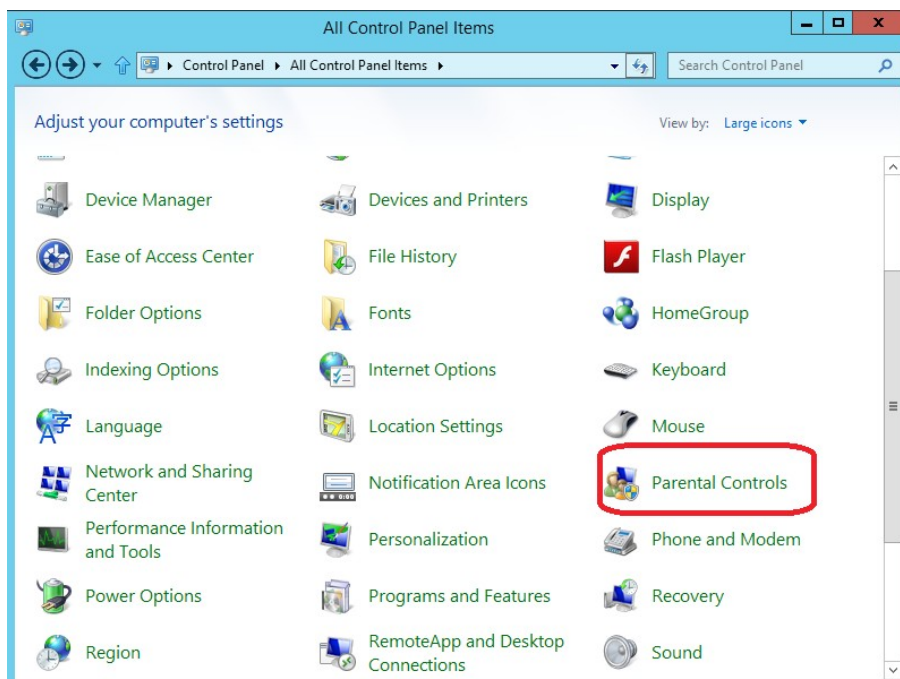


Figura 3: Painel de Controle do Microsoft Windows 8 e 8.1 e respectivos arquivos CPL (Control Panel Applet)



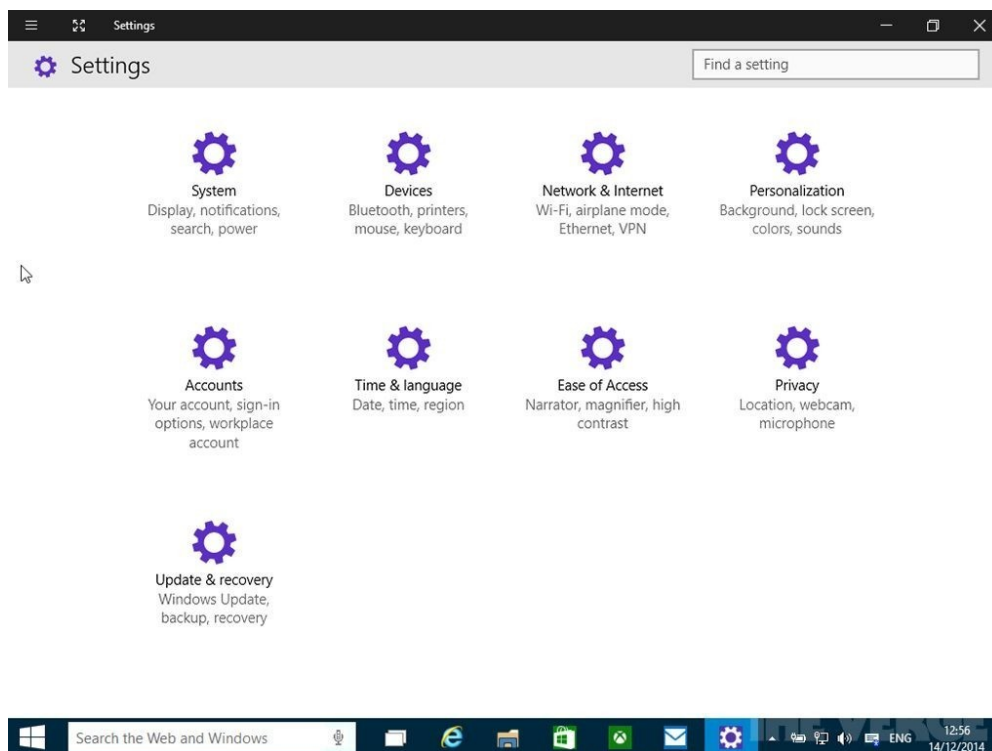


Figura 4: Painel de Controle do Microsoft Windows 10 e respectivos arquivos CPL (Control Panel Applet)

Estrutura

Um arquivo CPL é uma DLL (biblioteca de vínculo dinâmico) ou um EXE e por isso herda toda a estrutura já definida para o formato PE (Portable Executable). A diferença mais notável é que os arquivos CPL normalmente possuem somente uma função exportada chamada `CPIApplet()`, enquanto as DLL (biblioteca de vínculo dinâmico) em geral possuem várias funções exportadas (disponíveis para serem chamadas a partir de outro programa).

Execução

Quando a janela do Painel de Controle é aberta, todos os arquivos CPL na pasta de sistema do Windows são lidos e seus applets são carregados e exibidos na janela do Painel de Controle. Cada arquivo CPL pode conter um ou mais applets. É possível verificar isso com os métodos de execução abaixo:

Usando o `control.exe`

O programa `control.exe` (que carrega o Painel de Controle) utiliza a seguinte sintaxe para carregar applets CPL:



```
control.exe arquivo.cpl,@n,t
```

Onde n é o índice do applet dentro do arquivo, começando em 0 (zero) e t é o índice da aba deste applet, para applets com mais de uma aba, também começando em 0 (zero). Por exemplo, para carregar o primeiro applet do arquivo `main.cpl` posicionado na primeira aba, pode-se comandar:

```
C:\> control main.cpl,@0,0
```

Para ver a segunda aba deste mesmo applet, basta:

```
C:\> control desk.cpl,@0,1
```

Já para ver o segundo applet:

```
C:\> control main.cpl,@1,0
```

Caso o índice de applet ou aba seja inválido ou omitido, o Windows busca o applet padrão do arquivo e o exibe sem emitir erros.

Utilizando a função Control_RunDLL

Há ainda a possibilidade de se executar um arquivo CPL utilizando diretamente uma função da `shell32.dll` chamada `Control_RunDLL`. Este é o método que o Windows utiliza internamente e também é o que ocorre quando é dado um duplo-clique num arquivo CPL. Exemplo:

```
C:\> rundll32 shell32.dll,Control_RunDLL timedate.cpl,@0,0
```

O comando acima abre as propriedades de data e hora do Windows diretamente. Por ter essa característica, arquivos CPL podem ser executados diretamente no sistema operacional, bastando para isso um duplo-clique neles. Sendo assim, o efeito é similar aos arquivos EXE e à essa característica se deve a fama de que arquivos CPL são "DLLs que executam com dois cliques".

Utilizando o objeto Shell.Application

Um script em VBScript ou jscript pode carregar um CPL através do método `ControlPanelItem()` de um objeto `Shell.Application`. O código abaixo carrega o applet de configuração de joysticks:

VBScript

```
Dim obj
Set obj = CreateObject("Shell.Application")
```



```
obj.ControlPanelItem("joy.cpl")
```

Algumas relações de Applets nativos do Sistema Operacional Windows:

Windows 3.x

Nome do arquivo	Nome do ícone no Painel de Controle
CPWIN386.CPL	Avançado
DRIVERS.CPL	Controladores
MAIN.CPL	Cores, Fontes, Portas, Mouse, Área de Trabalho, Teclado, Impressoras, Internacional, Data e Hora e Rede (se instalada)
SND.CPL	Som

Windows XP

Nome do arquivo	Nome do ícone no Painel de Controle
Access.cpl	Ações de acessibilidade
Appwiz.cpl	Adicionar ou remover programas
Desk.cpl	Vídeo
Hdwwiz.cpl	Adicionar hardware
Inetcpl.cpl	Opções da Internet
Intl.cpl	Opções regionais e de idiomas
Irprops.cpl	Dispositivos Infra-vermelho
Joy.cpl	Controladores de jogo
Main.cpl	Mouse
Mmsys.cpl	Sons e dispositivos de áudio
Ncpa.cpl	Conexões de rede
Nusrmgr.cpl	Contas de usuário
Nwc.cpl	Serviços de Gateway para NetWare
Odbc32.cpl	Fontes de dados (ODBC), dentro de "Ferramentas administrativas"
Powercfg.cpl	Opções de energia
Sapi.cpl	Fala
Sysdm.cpl	Sistema
Telephon.cpl	Opções de telefone e modem



Timedate.cpl	Data e hora

Exemplos de ID Mitre CVE da Vulnerabilidade:

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0096>

1.2 O que são e como funcionam CPL(Control Panel Applet) Attacks:

Por ser um formato flexível e que pode ser executado com um duplo-clique, criadores de ameaças (atacantes) também criam malwares utilizando o formato CPL.

É comum no Brasil usuários receberem e-mails falsos com links para downloads de arquivos comprimidos (compactado em formato .RAR ou .ZIP) com um mais arquivos CPL dentro. Em geral, este tipo de arquivo não é transferido via rede com frequência pois seu maior uso está no Windows em si, portanto muito cuidado é necessário antes de dar um duplo clique em arquivos CPL recebidos por e-mail ou que tenham sido baixados na internet.

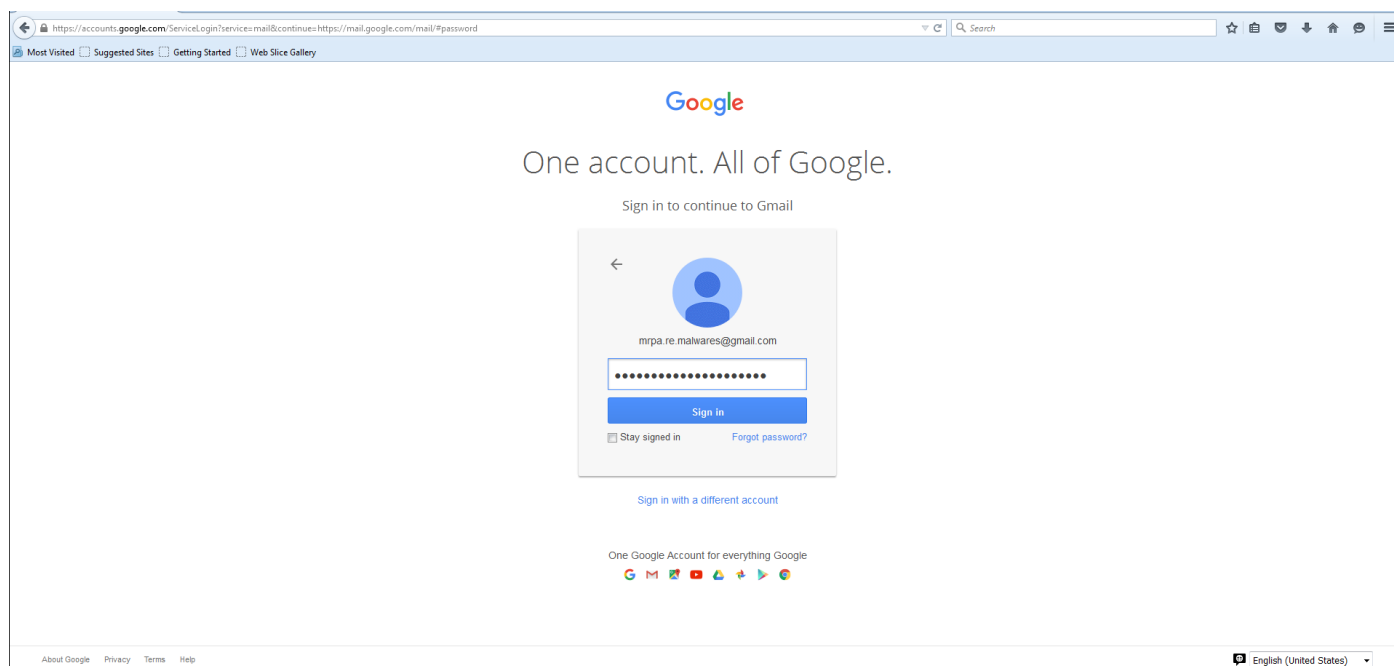


Figura 5: Email criado no SIS Labs para validar o testes de vulnerabilidades e exploração



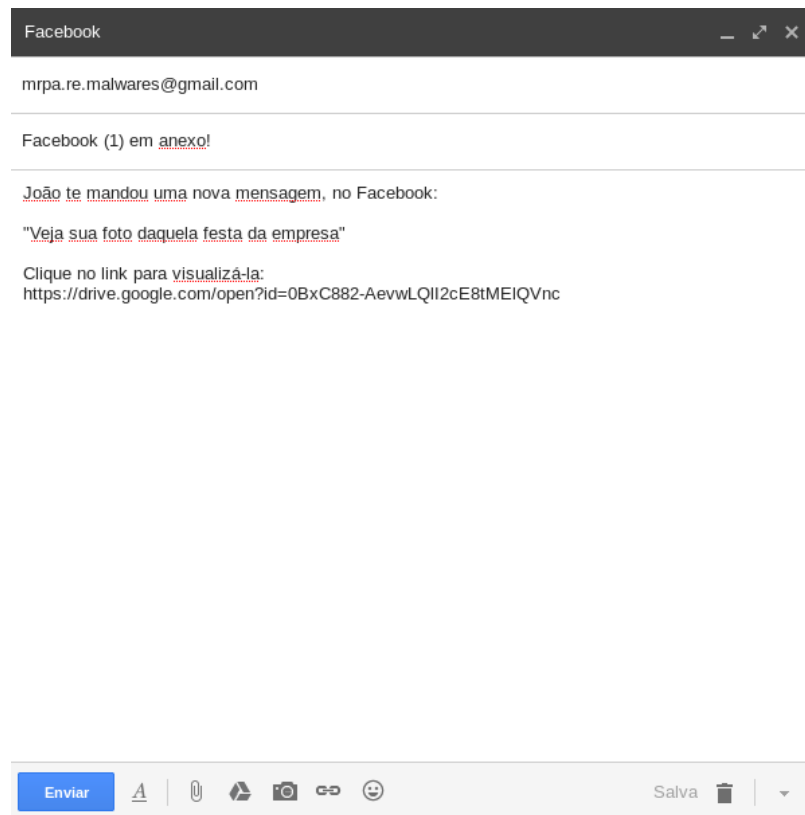


Figura 6: Email criado no SIS Labs para com texto e link para Google Drive (com arquivo .CPL) para realização dos testes de vulnerabilidades e exploração.



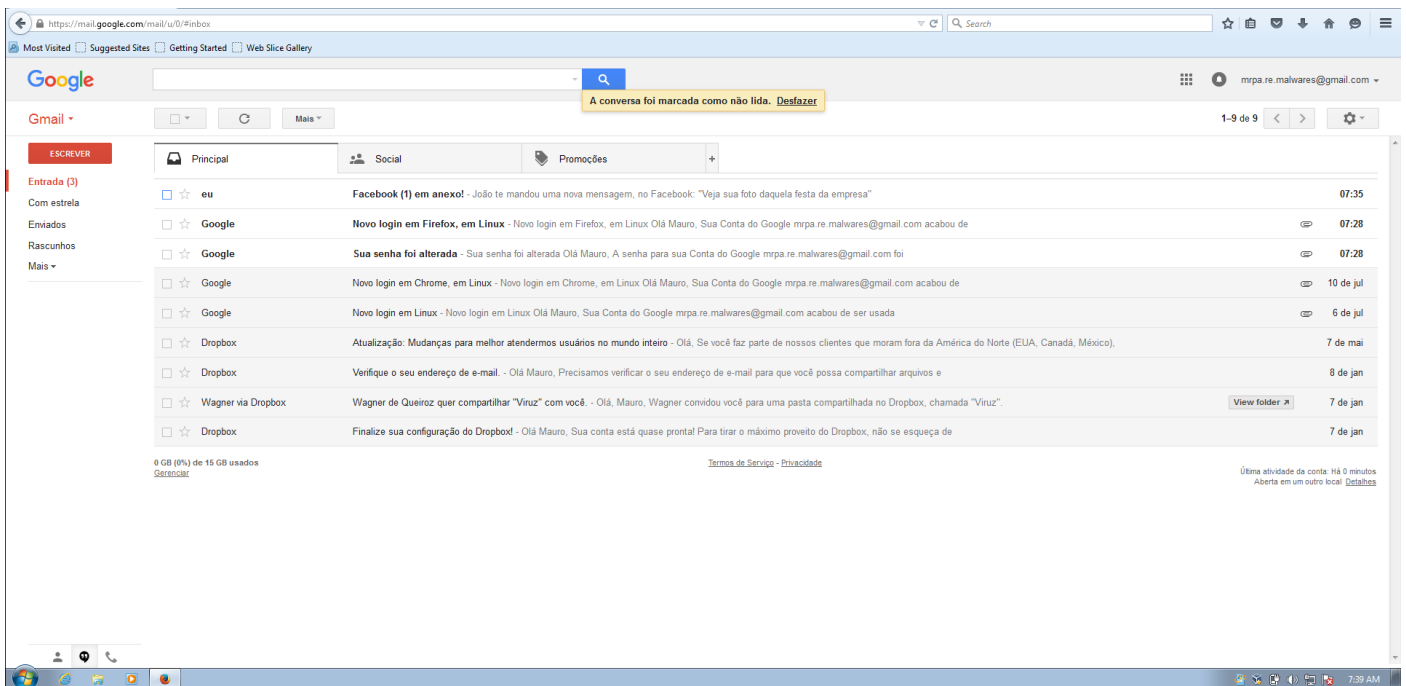


Figura 7: O email para testes. Não é considerado spam pelo Gmail.

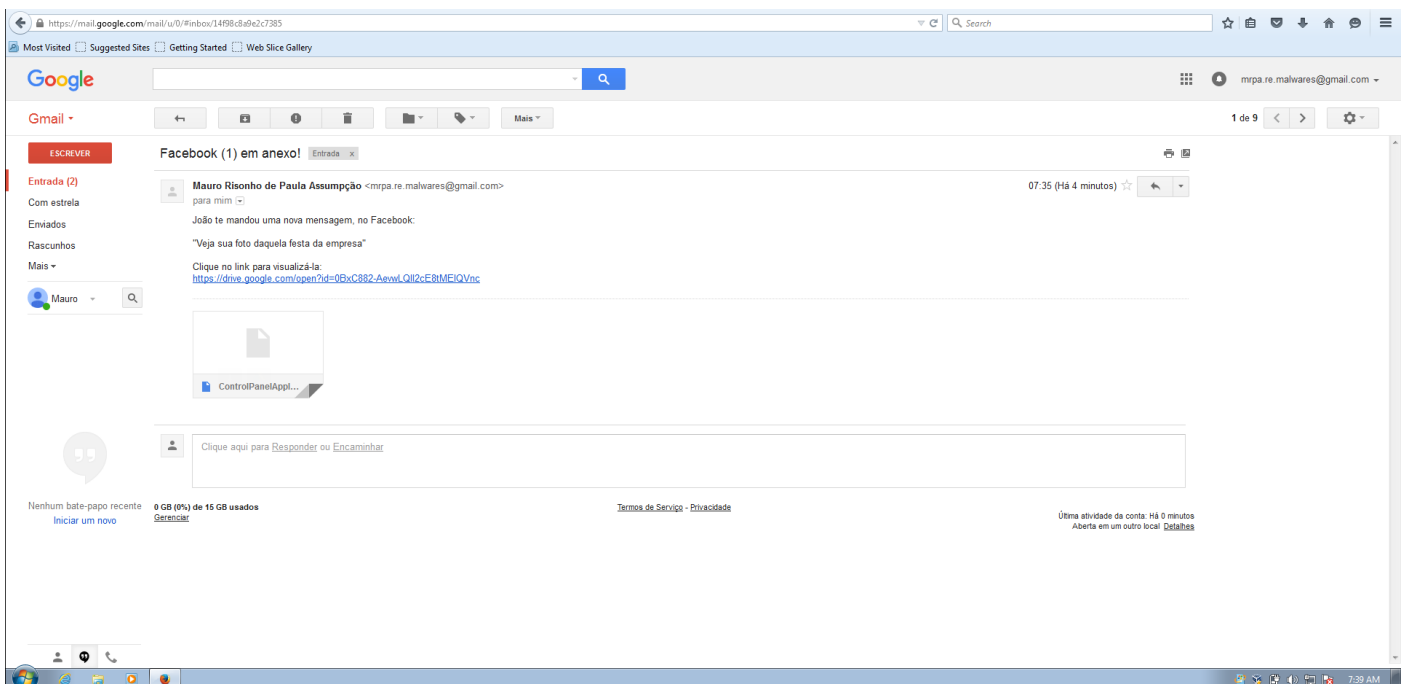


Figura 8: O email com link para Google Drive contendo o arquivo CPL (Control Panel Applet) não é bloqueado pelo Gmail.



Na figura 6 pode ser visto um exemplo de e-mail falso (phishing) com um link para um malware em CPL, na época catalogado como TROJ_BANLOAD ou TROJ_CHEPRO. Por ser um tipo de arquivo pouco conhecido, mesmo por analistas de segurança; vários softwares de proxy e outras soluções de segurança não tratam o download de um arquivo CPL com a devida atenção, imaginando ser um componente legítimo e sem nenhuma ameaça ao sistema operacional Windows.

O bloqueio de download de arquivos deste tipo não deve ser tão rígido, no entanto, porque alguns fabricantes de hardware distribuem arquivos CPL juntamente com os programas e drivers de dispositivo, mas deve ter alguma solução em APT (Advanced Persistent Threat) que mitigue este tipo de ataque.

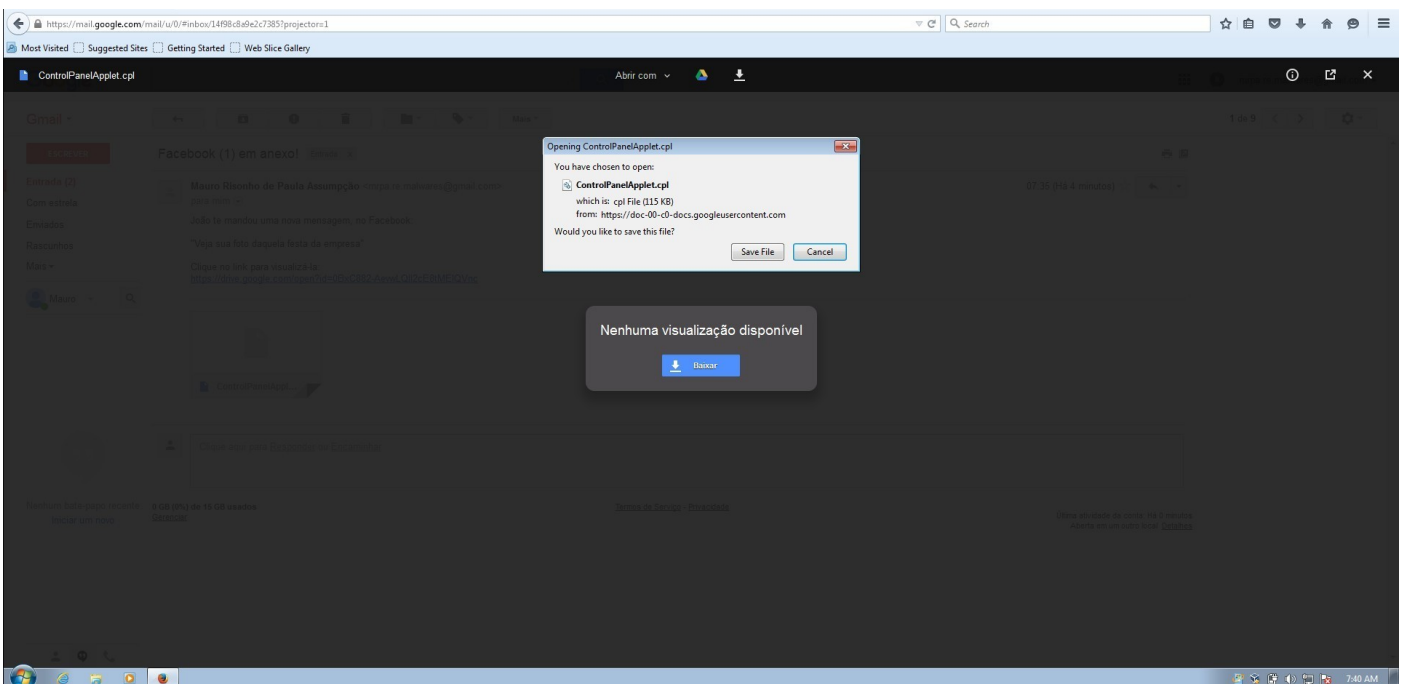


Figura 9: Praticamente nenhum antivírus tradicional irá detectar o arquivo CPL (Control Panel Applet) como ameaçador, pois é um componente inofensivo do Windows a princípio.



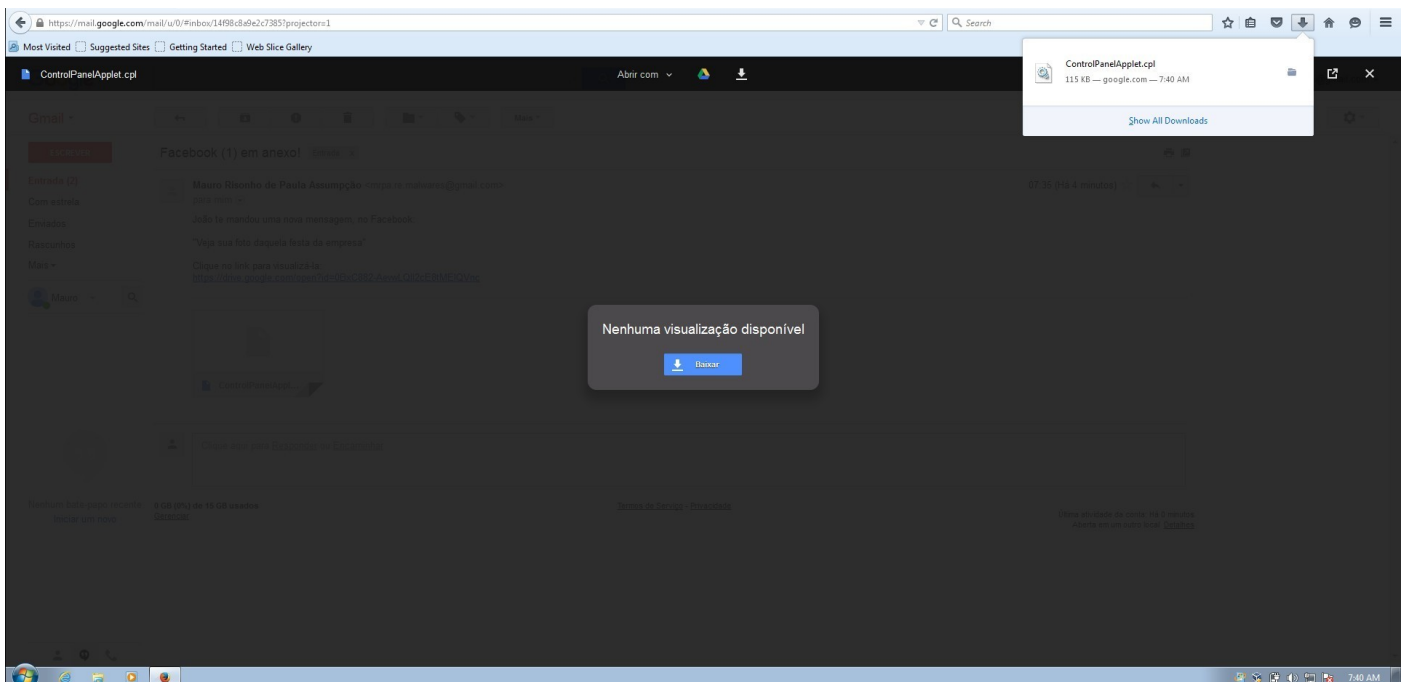


Figura 10: O download do CPL (Control Panel Applet) é feito normalmente pelo browser

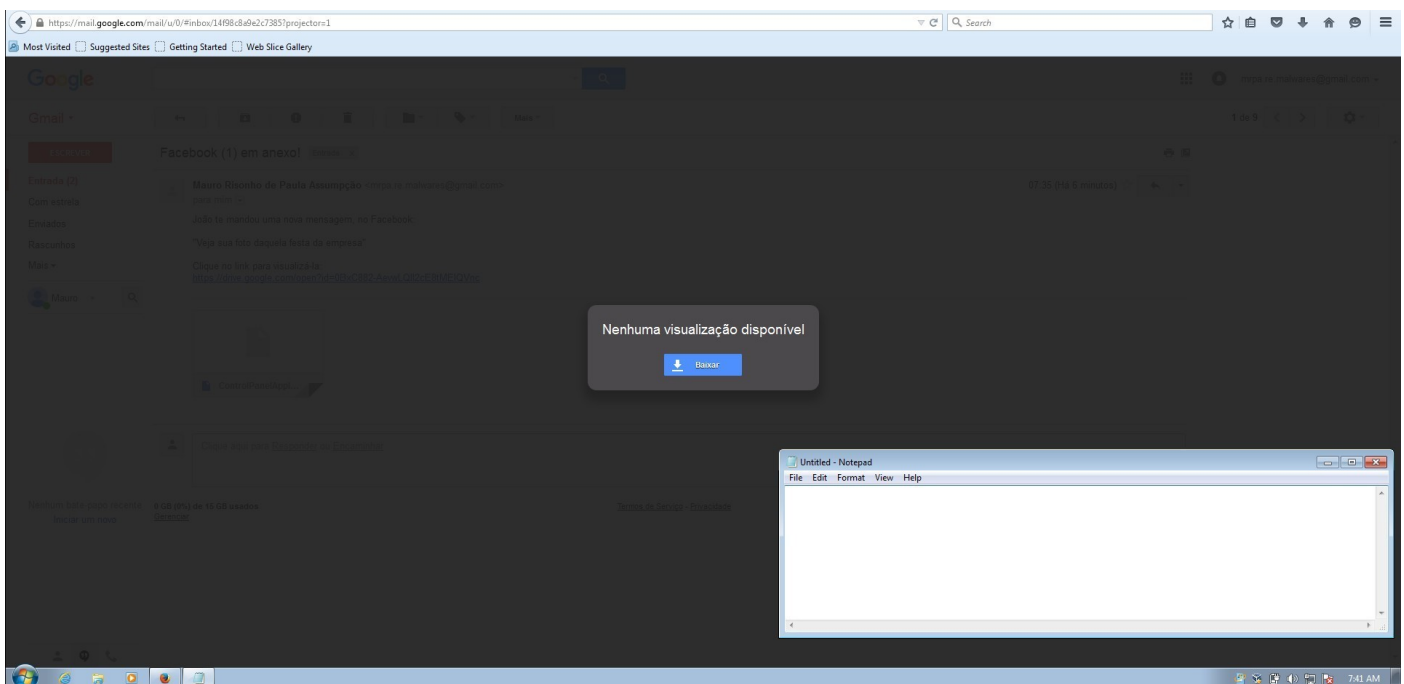



Figura 11: O usuário clica 2 vezes no arquivo CPL (Control Panel Applet) e o notepad é executado sem restrições, o que prova que poderia ser executado um arquivo malicioso qualquer e ganhado acesso administrativo ao Windows.



Virus total do Sampler

<https://www.virustotal.com/pt/file/eb4a382b2497fd813e27cf8b059f204e1a9302c4f2244beb35310be0c5b55dd4/analysis/1441825645/>

Comunidade
Estatísticas
Documentação
Dúvidas
Sobre
Português
Junte-se à comunidade
Entrar




SHA256: eb4a382b2497fd813e27cf8b059f204e1a9302c4f2244beb35310be0c5b55dd4

Nome do arquivo: ControlPanelApplet.cpl

Taxa de detecção: 1 / 56

Data da análise: 2015-09-09 19:07:25 UTC (0 minutos atrás)



Análise

[File detail](#)
[Informações adicionais](#)
[Comentários](#)
[Votos](#)

Antivírus	Resultado	Atualização
CAT-QuickHeal	(Suspicious) - DNAScan	20150909
ALYac	✔	20150909
AVG	✔	20150909
AVware	✔	20150901
Ad-Aware	✔	20150909
AegisLab	✔	20150909
Agnitum	✔	20150909
AhnLab-V3	✔	20150909
Alibaba	✔	20150902
Antiy-AVL	✔	20150909
Arcabit	✔	20150909
Avast	✔	20150909
Avira	✔	20150909

Figura 12: 1 de 56 fabricantes de AntiVirus, detectou como um sampler suspeito, mesmo sendo um teste



1.3 Processo Funcional de Ataques por arquivos .CPL:

Por ser um formato flexível e que pode ser executado com um duplo-clique, criadores de ameaças (atacantes) também criam mal

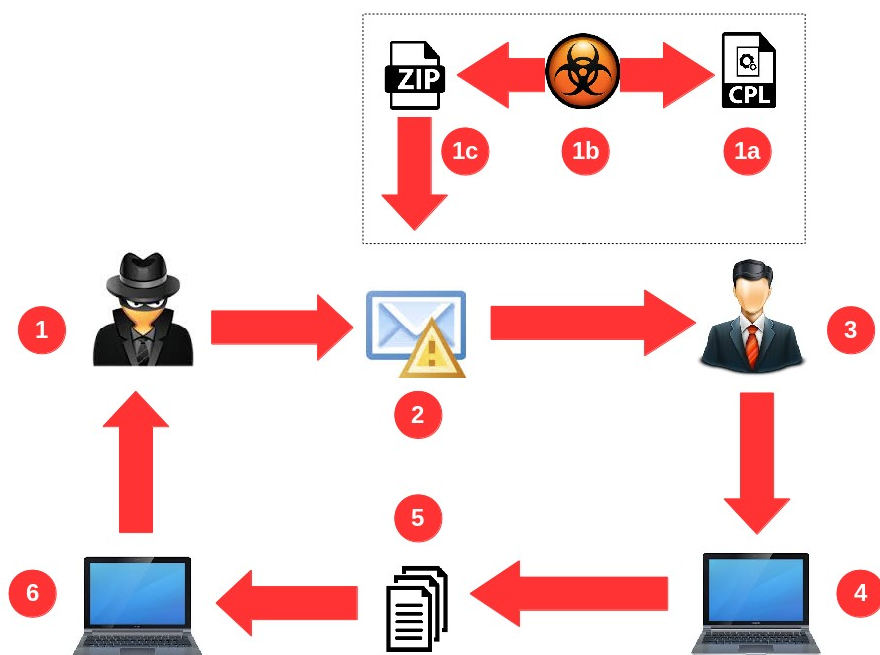


Figura 13: Processo de Ataques por arquivos .CPL (Control Panel Applet)

Sequencia do Ataque:

N.	Descrição (Passo-a-Passo)
1	O atacante escolhe os alvos a serem atacados
1A	Através de phishings, o atacante anexa um executável malicioso, em formato de arquivo CPL (Control Painel File)
1B	Arquivo malicioso é inserido no arquivo CPL
1C	Arquivo CPL com arquivo malicioso, é inserido dentro do arquivo *.ZIP.
2	Email Phising é enviado para a vítima/alvo.
3	O usuário clica no email e abre o arquivo *.ZIP, com CPL e executa automaticamente o

	arquivo malicioso
4	O arquivo malicioso, é executado pelo Painel de Controle do Windows do usuário alvo e o atacante ganha acesso e controle total da máquina.
5	O atacante com o controle total da máquina/alvo, passa a roubar todas as informações da máquina/alvo.
6	O atacante tem controle total da máquina/alvo, pois muitos AV, não tomam ações de bloqueio ou remoção, por identificarem com um arquivo benéfico e legítimo, afinal é um arquivo usado pelo Windows (CPL) em alguns casos, obter acesso via outro ataque na sequência por BufferOverflow ¹ .

Para demonstração desse tipo de ataque, foi criado nos Labs do SIS, da Agility Networks, um sampler como PoC (Proof of Concept) em ambiente seguro e controlado. Um sampler neste caso em específico, é nada mais que um projeto feito em Visual Studio Community 2015, usando a linguagem C++, compilado em formato de arquivo .DLL e renomeado para .CPL, justamente para criação de um arquivo do sistema operacional Windows, comumente usado no Painel de Controle.

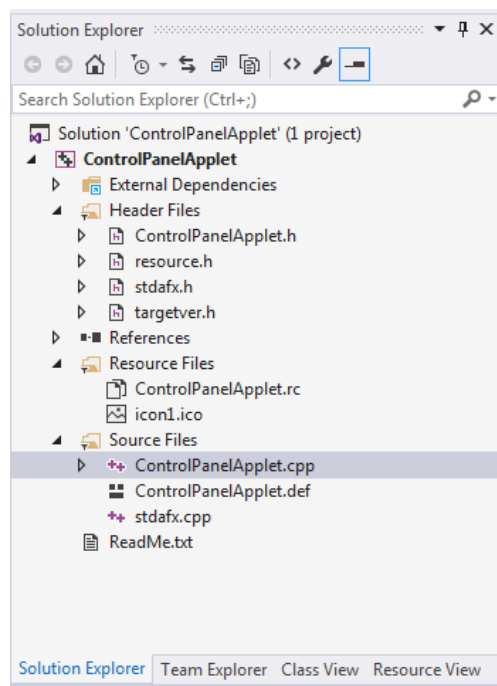


Figura 14: Projeto para geração de um sampler controlado em Laboratório (SIS Labs), provando a possibilidade de criação de arquivos no formato CPL (Painel de Controle do Windows).



Dentro de arquivo .CPL, criamos código-fonte para executar o Notepad, quando o usuário clicar 2 vezes no mesmo arquivo, provando que é executável e que a mesma técnica poderia ser usada num ataque mais sofisticado, que seria o CPL Attack.

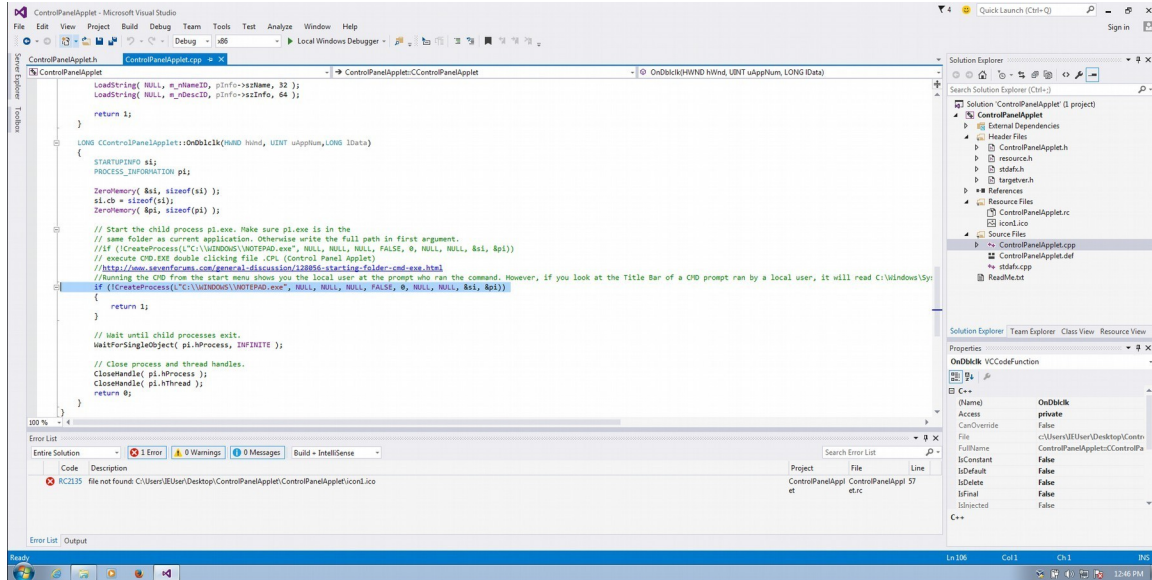


Figura 15: Fragmento do projeto do sampler, onde contém a função para execução.

Abaixo segue fragmento de código-fonte (na linguagem C++) criado em Microsoft Visual Studio Community 2015 (<https://www.visualstudio.com/pt-br/downloads/download-visual-studio-vs.aspx>), com a referência em vermelho, que é onde está o código-fonte que poderia executar comandos e aplicações do sistema operacional Windows.

```

ControlPanelApplet.cpp

// 31/08/2015 09:00 Brazil
// Author: carlosmsr (https://www.codeproject.com/Members/carlosmsr)
// Forked: Mauro Risonho de Paula Assumpcao aka firebits
// Sampler for Palo Alto Traps - PoC
// Sampler CPL Attacks (Malwares)
// ControlPanelApplet.cpp : Defines the exported functions for the DLL application.
// output ControlPanelApplet.dll
// rename ControlPanelApplet.cpl (Control Panel Applet) for Control Panel Windows
// = Start (Button) -> Control Panel Windows
// issues send email for mauro.assumpcao@agilitynetworks.com.br or
// mauro.risonho@gmail.com
// created Windows 7 64bits and Microsoft Visual Studio Community 2015
// https://www.codeproject.com/articles/29264/WebControls/
//
// LICENSE:
This article, along with any associated source code and files, is licensed under
The Code Project Open License (CPOL) http://www.codeproject.com/info/cpol10.aspx
    
```



```
#include "stdafx.h"
#include <windows.h>
#include <stdio.h>
#include "ControlPanelApplet.h"

namespace ControlPanelApplet
{
    CControlPanelApplet* CControlPanelApplet::m_pThis = NULL;

    CControlPanelApplet::CControlPanelApplet ()
    {
        m_pThis = this;
    }

    // Callback members
    LONG APIENTRY CControlPanelApplet::CPlApplet (HWND hWnd, UINT uMsg, LONG
lParam1, LONG lParam2)
    {
        CControlPanelApplet* pApplet = m_pThis;

        switch (uMsg)
        {
            case CPL_DBLCLK:
                return pApplet->OnDblclk(hWnd, lParam1, lParam2);

            case CPL_EXIT:
                return 0;

            case CPL_GETCOUNT:
                return 1;

            case CPL_INIT:
                return 1;
        }
    }
}
```




```
case CPL_INQUIRE:
    return pApplet->OnInquire(lParam1, (CPLINFO*)lParam2);

case CPL_NEWINQUIRE:
    return pApplet->OnNewInquire(lParam1, (NEWCPLINFO*)lParam2);

case CPL_STOP:
    return 1;

case CPL_STARTWPARMS:
    return pApplet->OnDblclk(hWnd, lParam1, lParam2);

default:
    break;
}
return 1;
}

// Default command handlers
LONG CControlPanelApplet::OnInquire(UINT uAppNum, CPLINFO* pInfo)
{
    pInfo->idIcon = IDI_ICON1;
    pInfo->lData = 0;
    pInfo->idName = m_nNameID;
    pInfo->idInfo = m_nDescID;

    return 0;
}

LONG CControlPanelApplet::OnNewInquire(UINT uAppNum, NEWCPLINFO* pInfo)
{
    pInfo->dwSize = (DWORD)sizeof(NEWCPLINFO);
    pInfo->dwFlags = 0;
    pInfo->dwHelpContext = 0;
}
```



```
pInfo->lData = 0;
pInfo->szHelpFile[ 0 ] = '\\0';

LoadString( NULL, m_nNameID, pInfo->szName, 32 );
LoadString( NULL, m_nDescID, pInfo->szInfo, 64 );

return 1;
}

LONG CControlPanelApplet::OnDblclk(HWND hWnd, UINT uAppNum, LONG lData)
{
    STARTUPINFO si;
    PROCESS_INFORMATION pi;

    ZeroMemory( &si, sizeof(si) );
    si.cb = sizeof(si);
    ZeroMemory( &pi, sizeof(pi) );

    // Start the child process p1.exe. Make sure p1.exe is in the
    // same folder as current application. Otherwise write the full path
in first argument.
    //if (!CreateProcess(L"C:\\WINDOWS\\NOTEPAD.exe", NULL, NULL, NULL,
FALSE, 0, NULL, NULL, &si, &pi))
        // execute CMD.EXE double clicking file .CPL (Control Panel Applet)
        //http://www.sevenforums.com/general-discussion/128056-starting-
folder-cmd-exe.html

        //Running the CMD from the start menu shows you the local user at the
prompt who ran the command. However, if you look at the Title Bar of a CMD prompt
ran by a local user, it will read C:\\Windows\\System32\\cmd.exe, showing that the
program is still being run from C:\\Windows\\System32.
        CreateProcess(L"C:\\WINDOWS\\NOTEPAD.exe", NULL, NULL, NULL,
FALSE, 0, NULL, NULL, &si, &pi)
    {
        return 1;
    }

    // Wait until child processes exit.
```



```
        WaitForSingleObject( pi.hProcess, INFINITE );

        // Close process and thread handles.
        CloseHandle( pi.hProcess );
        CloseHandle( pi.hThread );
        return 0;
    }
}
```

Source Code <https://www.codeproject.com/articles/29264/WebControls/>

A função CreateProcess cria um novo processo, que é executado independentemente do processo de criação. No entanto, para manter a simplicidade, a relação é referida como uma relação pai-filho (parent-child relationship), conforme <https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425%28v=vs.85%29.aspx>

Após a compilação com sucesso, vamos até a pasta onde está o arquivo em formato .DLL, qual iremos renomear para .CPL, que será reconhecido pelo Sistema Operacional Windows, como parte integrante do Painel de Controle do Windows.

Vamos executar o arquivo .CPL, clicando 2 vezes com o botão esquerdo do mouse. Automaticamente será executado o Notepad.exe, o que não é esperado pelo usuário. Tudo indica, no caso que se fosse algum artefato malicioso, seria executado e não detectado por Antivirus tradicionais.

Isso se deve pelo atacante já conhecer o sistema operacional e executar comandos/aplicações nativas. Mas atacantes mais experientes poderiam efetuar ataques mais avançados, que vão até o kernel do Windows.



1.4 Testes após execução e compilação

Será necessário baixar o pacote System Internals (<https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>), para usarmos um aplicativo chamado Process Explorer e acompanhar os processos e chamadas do Windows, e outras partes do Sistema Operacional, para execução do sampler:

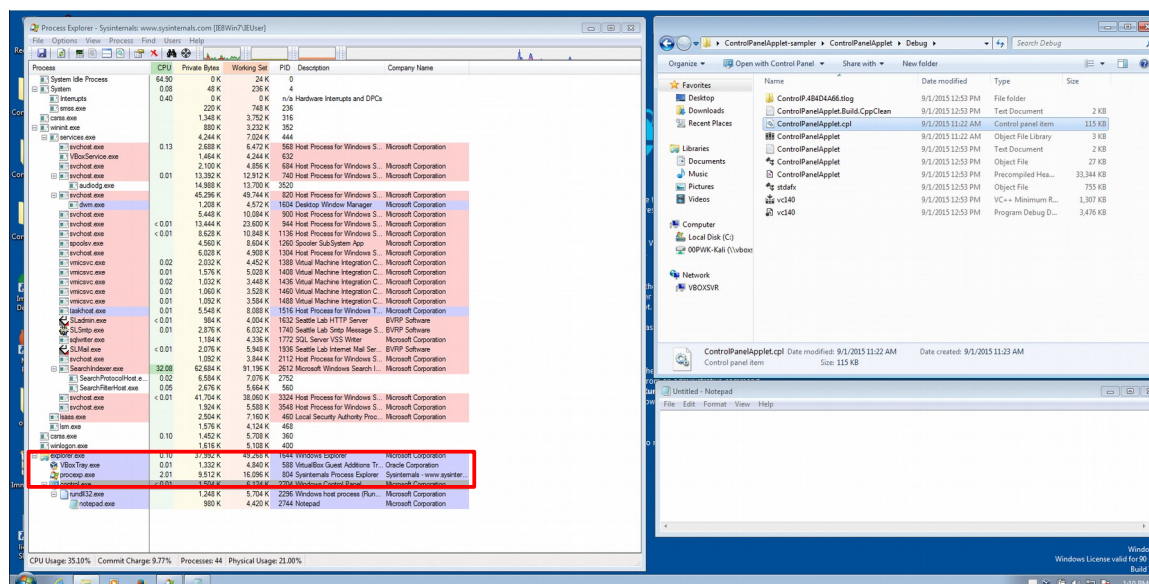


Figura 16: O sampler em execução, através do Painel de Controle do Windows (Control.exe).

Repare que ao usuário clicar no arquivo .CPL, o Windows, automaticamente abre o executável Control.exe, faz chamada para rundll32.exe e depois executa Notepad.exe. Após a execução do notepad.exe, o Control.exe é executado por alguns segundos e o Control.exe é “matado ou kill”, sem matar os processos filhos, como por exemplo, rundll32.exe e o próprio notepad.exe, neste cenário com processxp.exe, do System Internals.

Uma ação não esperada para usuário, pois ele queria abrir arquivo de foto, conforme o Phishing que direcionaram à ele.



¹ BufferOverFlow = uma anomalia onde um programa, ao escrever dados em um *buffer*, ultrapassa os limites do *buffer* e sobrescreve a memória adjacente. Esse é um caso especial de violação de segurança de memória. BufferOverFlow podem ser disparados por entradas que são projetadas para executar código, ou alterar o modo como o programa funciona. Isso pode resultar em comportamento errados do programa, incluindo erros de acesso à memória, resultados incorretos, parada total do sistema, ou uma brecha num sistema de segurança, para ganhar acesso irrestrito em alguns cenários. Portanto, eles são a base de muitas vulnerabilidades de *software* e podem ser explorados maliciosamente, remotamente ou localmente, conforme o caso.

Conclusão:

Conforme já visto com um simples email com um arquivo .ZIP anexado, contendo um arquivo .CPL, pode simplesmente “passar” por várias camadas de segurança de rede e de aplicação, pois muitos software e hardwares de segurança, interpretam como arquivo seguro e sem ameaças. A melhor forma de se prevenir contra esta vulnerabilidade é ter equipamentos que tenham funções de proteção por hardware e por software disponibilizado por cada fabricante, contra APT (Advanced Persistent Threat).

Esse tipo de Ataque, que é o CPL Attack, pode comprometer grandes empresas, instituições financeiras e outros segmentos, caso não possuam nenhuma solução contra APT(Advanced Persistent Threat).

Agradecimentos:

- Paulo Elias Junior - <https://br.linkedin.com/in/pauloeliasjr>

Agility Networks, IT Project Analyst – PROJETOS

Referencias:

<https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>

<https://support.microsoft.com/pt-br/kb/192806>

<http://researchcenter.paloaltonetworks.com/2015/03/palo-alto-networks-traps-prevents-exploitation-of-cve-2010-2568cve-2015-0096-stuxnet-zero-day/>

<http://blog.trendmicro.com/trendlabs-security-intelligence/control-panel-files-used-as-malicious-attachments/>

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0096>

<https://www.codeproject.com/articles/29264/WebControls/>

