1. Introducere:

In cadrul acestui curs, vom discuta despre Chamois cel mai de impact Botnet din 2018. Prezentarea originală este făcută de Maddie Stone care este Senior Reverse Engineer şi Tech Lead pe Google Play Protect.

2.
Chamois este un Android Botnet care face ca anumite aplicaţii să realizeze anumite acţiuni frauduloase fără ca acestea să ştie.
Acţiunile frauduloase sunt:

- Ad fraud - click-uri false care aduc un comision
- SMS fraud - trimiterea de SMS-uri premium, adica cu costuri mari. Costurile sunt plătite ulterior pe factura de la abonament. Acesta este un lucru obişnuit la noi şi în ţările din est dar nu şi în America.
- Install fraud - Instalari false care aduc un comision

Dacă vreţi să intraţi în mai multe detalii am adăugat linkuri pentru fiecare tip de acţiune.

3.
Ce este un Botnet?
Un botnet este o reţea de maşini infectate care sunt coordonate de un centru de comandă şi control. Acesta se prescurtează C&C sau C2. Puterea unui botnet stă în puterea de calcul şi cantitatea de stocare distribuită pe care o are. Cu cat mai multe maşini infectate cu atât puterea unui botnet este mai mare.

Backdoor?
O aplicaţie care permite executarea unor operaţii nedorite, cu un efect potenţial negativ, care pot fi controlate de la distanţă.

PHA?
Potentially Harmful Applications - sunt aplicaţii care ar putea să expună utilizatorul, datele utilizatorului sau dispozitivul la risc. Generic ele sunt referite ca **malware**.

Putem vedem un status clar al aplicaţiilor PHA pe PHA status.

Google Play Protect?
Un serviciu de protecţie împotriva ameninţărilor dispozitivelor mobile. Ce este important de spus este că Google Play Protect oferă siguranţă tuturor dispozitivelor certificate Google. Acesta identifică aplicaţii maliţioase chiar dacă acea aplicaţie nu a fost instalată de pe Google Play. Am pus nişte link-uri pentru mai multe detalii.

APK?
Scriem codul aplicaţiei şi adăugăm toate resursele necesare dar cum ajung acestea pe telefon. SDK-ul de Android compilează codul împreună cu toate datele şi resursele într-un APK, sau Android Package. Acesta este o arhivă cu extensia **.apk**. Un APK conţine tot ce este necesar pentru a aplicaţie Android şi este fişierul pe care dispozitivele Android îl folosesc pentru a instala aplicaţia.

Distribuţia aplicaţiei?
O aplicaţie Android poate fi distribuită pe mai multe canale:

- Principalul canal de distribuţie şi cel mai mare este Google Play

- Sideloaded - prin instalarea directă a unui fişier APK - care acesta poate fi obţinut prin mai multe metode: de pe un site web, trimis prin altă aplicaţie sau încărcat folosind ADB aşa cum face Android Studio
- Third Party Stores: Alte magazine de aplicaţii: Amazon, Baidu
- Pre-Installed, preinstalate: care sunt instalate de către producător direct în imaginea de sistem

4.
Un mic sumar despre Chamois:
- Ca şi categorie de de PHA Chamois este un Backdoor - deoarece permite execuţia unor operaţii cu efecte potenţial negative
- A fost detectat iniţial la mijlocul lui 2016
- Ca şi mod principal de distribuţie a fost printr-un SDK care dezvoltatorii 3rd party îl includeau în aplicaţiile lor, fără să ştie că este botnet backdoor
- Sunt 4 variante distincte
- Fiecare varianta are 4 până la 6 etape
- Ca şi payload sunt de 4 tipuri:
    - fraudă cu SMS premium
    - instalări false
    - fraudă cu reclame
    - încărcarea arbitrară de de module

5.
După cum am zis, varianta 1, a fost detectată prima dată la mijlocul lui 2016 pe Google Play. Ce este important de enunţat este că la acel timp Chamois era distribuit în mod principal pe Google Play.

După în noiembrie 2016, câteva luni mai târziu, a apărut varianta 2. Diferenţa între varianta 1 şi 2 este că în prima variantă se făcea doar Ad fraud iar în a doua s-a adăugat Premium SMS fraud. Varianta 2 se afla tot pe Google Play.

Apoi în martie 2017 Google a postat un blog în care spunea că au eliminat Chamois din Google Play, dar uite cum Timeline-ul nostru nu se termină în 2017.
După martie 2017 a urmat o perioadă de liniște în care nu s-a detectat nimic legat de Chamois. În noiembrie 2017 au început să apară noi semnături.
Abia în ianuarie 2018 două persoane diferite au reuşit să determine ca aceste semnături sunt ale Chamois.

Un lucru important de spus este că Google verifică toate imaginile de sistem ale OEMs înainte de lansare pentru breşe de securitate. Dacă vreo aplicaţie prezintă vreo problemă este trimisă pentru a fi verificată de o persoană. Într-unul din cazuri s-a găsit o aplicaţie foarte obfuscată, fără icon şi era numită Sales Tracker/Urmăritor de vânzări, dar fără să prezinte nimic evident. Între timp o altă persoană a observat traffic pentru reclame foarte ciudat. Corelând cele două au descoperit varianta 3 a Chamois. Aceasta era mult mai sofisticată şi adăuga încă 2 stagii faţă de varianta 1 şi 2 dar păstra aceleaşi payload-uri.

Având în vedere complexitatea problemei în vara anului 2018 s-a decis începerea unei investigaţii formată din mai multe echipe. Totodată a apărut şi varianta 4. Indiferent până în decembrie 2018 Chamois a fost adusă la stadiul de mentenanţă şi monitorizare ceea ce înseamnă că este sub control.

6.
Ok. De ce ”Cea mai de impact?”
Complexitate tehnică - o să revin mai tarziu peste detaliile asta amuzante

Ce este foarte interesant este că au folosit multiple canale de distribuţie şi multiple tehnici de distribuţie.
O modalitate prin care să blochezi infectarea cu malware în diferite ecosisteme este să studiezi cum ajunge acolo şi cum se înmulţeşte.
Un lucru ce ia ajutat foarte mult a fost că odată ce un canal de distribuţie era oprit alte apăreau având în vedere că aveau backup şi o mulţime de metode de a se răspândi.

Trecând peste acestea, Chamois este un produs al unei echipe de ingineri. Această echipă de ingineri are procese bine definite/mature de lansare, de dezvoltare şi testare.
Cel mai important este că actorul din spatele acestora are finanţare şi are o infrastructură sofisticată de lansare şi testare.

Şi nu în ultimul rând folosesc diferite tehnici de fraudă cu reclame care sunt ascunse în diferite părţi ale ecosistemului astfel încât dacă una nu este funcţională, monetizarea botnetului nu oprită complet.

8.
Deci să trecem la partea amuzantă.

După cum am prezentat au existat 4 variante. În continuare ne vom focusa pe varianta 3 şi 4 având în vedere că acestea sunt cele mai avansate tehnic.

9.
Dar să mergem puţin înapoi la variantele din 2016 care în anul acela erau deja destult de sofisticate având în vedere complexitatea sistemului de securitate Android. Un lucru evident este că nu căutat să prindă vreun lucru uşor de accesat.

Au combinant APK-uri dezarhivate dintr-un format propriu de arhivă criptat într-un mod propriu. Au combinat fişiere JAR diferite împreună cu cod nativ(C++) pentru a realiza anumite comportamente. După prezentare au venit în forţă cu un sistem în 6 stagii în loc de 4.

10.
Ok. Deci au venit cu sistemul acesta complex pentru variantele 3 & 4. stagiu = etapa

Primul lucru pe care îl văd companiile de antivirus şi Google sunt APK-urile. Acestea de obicei au o semnătură(un hash corespunzător APK-ului) şi sunt folosite pentru a le detecta pe dispozitive.

Totuşi în Etapa 1 al Chamois nu se întâmplă mai nimic. Acesta are doar rolul de a se dezarhiva. Acum, Etapa 2 şi toate etapele care vin după pot fi categorizate astfel:

- Chamois loader - cel care încarcă următoarele etape. Acesta include tehnici de analiză şi ofuscare super sofisticate.
- Frameworkul Chamois - aceasta este partea în care lucrurile legate de Backdoor au loc. Aici intră în joc etapa 4 care va trimite apeluri către infrastructura de comandă şi control(C2) şi va întreba dacă framework-ul are nevoie de vreun update. Dacă răspunsul este pozitiv, etapa 5 este updated, dacă răspunsul este negativ va folosi

stagiul pe care îl are arhiva specială. Etapa 5 este responsabil pentru payload-urile diferite.

- Acum urmează secţiunea de malicious payloads din etapa 6. Acestea sunt împărţite în 6a, 6b şi 6c. 6b este payload-ul maliţios. Acesta conţine peste 15 payload-uri maliţioase care încă sunt în dezvoltare. Etapele 6a şi 6c există pentru a susţine 6b în executarea comportamentului maliţios. Dacă este nevoie de cod native se apelează la etapa 6a(gen webview modification) iar dacă este nevoie de ceva legat de Android se apelează la etapa 6c pentru a putea executa nişte Ad fraud.

ELF = Android uses ELF .so(shared object) libraries for Java Native Interface. With Android Runtime(ART), the default since Android 5.0(Lollipop), all aplications are compiled into native ELF binaries on instalation

11.
Deci am spus câteva lucruri despre formatul de arhivă custom/personalizat, de unde provin majoritatea etapelor 3 plus fiind dezarhivate din fiecare etapă anterioară.

Acesta este cel mai stabil lucru legat de Chamois încă de la varianta 1 şi pană la varianta 4. Este similară cu o arhivă de tipul **zip** doar că tot ce ţinea de zip a fost schimbat.

Gândiţi-vă la formatul de fişier ca unul de tip Inception, prin includerea arhivelor una în alta.

Ce este foarte inteligent la acest tip de arhivă este că dacă aveţi o copie a formatului de arhivă personalizat, nu o puteţi, dezarhiva, cu excepţia cazului în care aveţi şi APK-ul din care a aparţinut. În plus dacă aveţi APK-ul dar nu aveţi formatul de arhivă nu puteţi despacheta sau afla care sunt etape 3-plus.

Prin urmare, făcând acest lucru şi că fiecare mostră de Chamois era independent generată a făcut ca acesta să fie mult mai greu de detectat şi de înţeles.

12.
Unul dintre cele mai dificile aspecte ale Chamois este tehnicile de anti-detectare cu adevărat sofisticate.
Chamois face un tip de debugging sau ofuscare la fiecare etapă, deci în toate cele 6 etape şi în variantele 3 şi 4.

Deci pentru etapele 1 şi 2 din fiecare eşantion/sample fiecare string, nume de clasă sau de fişiere este randomizat. Astfel încât nu se poate aplica verificarea de semnături deoarece nu au folosit, de exemplu un nume de clasă sau pachet, cu aceeaşi lungime.

Ok. Acum scopul etapei 3 este să facă anti-debugging. Dacă ne întoarcem înapoi la graficul care l-am prezentat, etapa 3, era ultima etapa din Chamois loader înainte ca framework-ul Chamois să fie încărcat. Astfel, acesta va completa doar dacă este foarte sigur că nu este analizat, depanat/depanat sau emulat.

Toate aceste lucruri despre tehnicile de anti-detecţie au fost prezentate de către Maddie la o conferinţă de securitate(Black Hat USA) în 2018. Le puteţi găsi în link-urile din prezentare dacă vreţi să aflaţi mai multe.

Dar ca şi rezumat, etapa 3 conţine in-place decryption, anti-reverse engineering, anti-emulare care conţine verificarea a peste 37 proprietăţi de sistem, verificarea architecturii

sistemului citind diferite fişiere din sistem, precum şi verificări folosind framework-ul Xposed(modificare comportament sistem) sau Monkey(crearea de eveniment clickuri, gesturi, evenimente de sistem).

Acesta este singurul mod în care ar continua, astfel încât nu ar încărca payload-urile, dacă exista vreo şansă ca acestea să fie analizate.

Făcând toate aceste lucruri şi prin faptul că fiecare eşantion de Chamois este generat independent, au făcut ca detecţia Chamois să fie foarte dificilă.

13.
Payload-urile sunt de 2 tipuri cele bune, din stanga, si cele rele, ascunse, cele din dreapta.

Modul prin care au reuşit să păcălească dezvoltatori să includă Chamois în aplicaţiile lor, a fost prin faptul că oferă o soluţie de plată pentru mobile. Având aceste tipuri de payloads care suportă diferite tipuri de plăţi au reuşit să îşi creeze o reputaţie sau o credibilitate pentru ei ca organizaţie. Aceste payload-uri bune nu prea au fost folosite. În general au suportat toate payload-urile maliţioase, care au făcut diferite tipuri de fraude cu reclame şi o mulţime de aceste tipuri, creşterea traficului şi trimiterea de SMS-uri premium.

14.
Unul dintre cele mai interesante părţi care arată cât de mult s-au gândit exact la ceea ce vreau să facă este că atunci când au decis să-şi lanseze payload-ul pentru premium SMS, au înţeles platforma Android, prin faptul că platforma va afişa în această avertizare în cazul în care este posibil să trimiteţi un SMS la un cod scurt care ar putea genera taxe pe factură.

Aşa că actorii Chamois au decis că nu este ok şi s-ar putea să îi dea de gol. Deci ce au făcut.

Dacă telefonul era rootat, adică aveau acces oriunde îşi adăugau o permisiune internă care de obicei nimeni nu o poate accesa. Ce face acea permisiune este să seteze că userul a bifat căsuţa cu Remember my choice şi a apăsat butonul Send. Această setare se făcea dinainte de a trimite un SMS pentru a evita aceste tipuri de avertizari.

Dacă nu era rootat. Nici o problema, foloseau Accessibility services. Accessibility services se foloseste pentru cei cu impaired vision/deficienţe de vedere. O făceau atât de repede încât utilizatorul nu ar fi apucat să o vadă.

15.
S-au gândit foarte atent la modul în care pot fi cei mai eficienţi şi să nu fie detectaţi, dar nu au făcut toate aceste lucruri şi le-au aruncat in the wild ca să vadă ce se întâmplă în speranţa că vor monetiza ceva. Au fost foarte deştepţi în modul de testare înainte de a implementa ceva nou.

Aşa cum am mai spus motoarele de AntiVirus în general văd doar etapa 1, fişierul APK, aşa că unul dintre primele lucruri pe le fac este să itereze ce fişiere şi string-uri includ în APK pentru a vedea care dintre semnăturile APK-urilor sunt descoperite de AV. Sistematic sa putut observa cum testeaza sistemul AV schimband toate proprietatile, modificand chiar si denumirele de iconite. Astfel puteau să detecteze ce reguli se foloseau în analiza Antivirusului şi pentru a ii face bypass.

Totodată folosesc servere de staging şi production astfel încât ei nu lansează totul şi pe langă aceasta ca orice echipă matură de ingineri folosesc Feature Flags.

Aşa că vor face disponibile anumite funcţionalităţi anumitor populaţii doar dacă ei permit asta cu ajutorul Feature Flags. Pentru asta se foloseau de locaţia ta geografică, pe ce dispozitiv se ruleaza şi ce provider de serviciu de telefonie mobilă foloseşti. Deci totul era controlat astfel încât dacă ceva nu ar funcţiona, ei ar putea să îl oprească oricând şi să îl şteargă din botnet înainte de a fi detectaţi.

16.
De asemenea, au o infrastructură de reţea destul de mare, deci când vorbim despre domenii de API C2, vorbim despre cum se face update, cum se face upgrade, ce payload-uri să folosească. Sunt cel puţin 10 din acestea.

Domeniile specifice pentru fiecare modul sunt cele care conţin payload-urile. Până acum s-au gasit cam mai mult de 20 de astfel de domenii.

Doar pentru partea de fraudă cu reclame s-au găsit peste 150 de domenii.

Cel mai interesant lucru despre partea aceasta este că ei nu generează nimic din acestea în mod manual. Ei lucrează cu furnizori mari de servicii de cloud şi folosesc servicii automate pentru a crea domenii noi.

17.
Un alt lucru interesant este Ce se întâmplă dacă eşti din China? Aşa că înainte de a încărca un payload se fac o mulţime de verificări pentru a determina dacă se poate să ruleze.

18.
Si acum lucruri amuzante. Tot ce am prezentat până acum nu au nici un rost dacă nu poţi să îl distribui, dacă nu poţi să adaugi BOTs în botnet. Ar fi totul pentru nimic.

Unul dintre cele mai mari căi de distribuţie a fost prin preinstalare convingând OEMs şi ODMs. Original Equipment Manufacturing(per his design), companiile care vând cipuri, Original Design Manufacturing(per anothers design) Designed by Apple.

Aşa că au continuat să convingă OEMs şi ODMs că,
avem un Sales tracker care te ajuta pe telefoanele cu marjă mică de profit,
o soluţie de plăți mobilă care suportă tipurile astea de plăţi,
un SDK pentru reclame(de fapt un SDK de monetizare).
Folosind toate acestea au reuşit să îi convingă să îl includă preinstalat pe dispozitive mobile pe care acestea le fabricau.

De asemenea l-au distribuit şi ca un SDK static, adică dezvoltatorilor. Astfel mulţi dintre ei au incluse în aplicaţiille lor acest SDK fără ca să ştie că este rău.

Un alt lucru interesant care nu a mai fost văzut la scară înainte este că s-au asociat cu alţi actori maliţioşi din ecosistemul Android pentru a fi distribuite ca ”plugin-uri”. Ce înseamnă aceasta este că alte aplicaţii şi alte SDK-uri dăunătoare ar descărca plugin-uri Chamois şi

le-ar rula în contextul aplicaţiei lor. Asta înseamnă că şi alţi dezvoltatori au fost implicaţi în distribuirea Chamois. Astfel au ajuns şi în alte familii de aplicaţii care descarcă conţinut maliţios.

Din cauza tuturor acestor aspecte a fost o luptă grea împotriva Chamois. Foarte multe metode de distribuţie.

19.
Unul dintre cele mai înfricoşătoare moduri de intrare în lanţul de aprovizionare a fost EagerFonts. Aceasta este o aplicaţie de font-uri care a fost inclusă întruna dintre platformele SoC. Această platformă SoC a fost convinsă prin faptul că, ”Uite sunt un dezvoltator de aplicaţie de fonturi şi o să permită utilizatorilor tăi cu aşa multe opţiuni prin care să vadă fonturile corect in fiecare limbă.”

Ce s-a întâmplat este că acel dezvoltator 3rd party a includ un SDK pentru reclame, care cred că deja ştiţi ce conţinea. Acest SDK făcea apeluri către un server remote şi făcea încărcare dinamică de cod pentru a descărca diferite plugin-uri şi să le ruleze în contextul aplicaţiei.

Unele dintre aceste plugin-uri au inclus Chamois, o altă familie cunoscută ca şi Snowfox(captură tastaturii, conexiuni remote, colectare informaţii sistem, descărcare/încărcare de fişiere, mai aducem nişte malware, DoS,...), precum şi altele.

Datorită faptului că s-a infiltrat în lanţul de distribuţie al platformei SoC-ului, a afectat peste 250 de OEM-uri diferite în peste 1000 de tipuri de device-uri.

După ce au fost contactaţi producătorii platformei au reacţionat imediat, au contactat clienţii şi au create un plan pentru a prevenii acest lucru în viitor. Au scos aplicaţia în aceeaşi zi.

21.
Lupta a fost una grea deoarece era împotriva unui sistem foarte sofisticat. Deci s-a demarat o investigaţie care a inclus mai multe echipe şi s-a început lupta pe mai multe căi. Odată cu informarea OEM în ceea ce priveşte oprirea aprovizionării şi distribuiţiei şi încheind cu prevenirea monetizării, precum şi cu prevenirea de noi instalări pe dispozitivele utilizatorilor şi blocarea infecţiilor existenet folosind Google Play Protect.

22.
Aşadar, aşa arată paşii făcuţi pentru a colabora cu OEMs:

1) Avertizarea OEMs the prezentă pe dispozitivele lor mobile
2) Obligarea acestora să facă un update prin OTA pentru a remedia problma Over the air
3) Post-mortem pentru a studia cum a ajuns problema pe dispozitiv
4) Plan de prevenţie în viitor

S-a creat un program de certificare prin care toate imaginile de sistem sunt trimise pentru a fi verificate înainte ca acestea să fie puse pe dispozitive.

23.
După cum vă puteţi da seama semnăturile din Google Play Protect nu era destul de multe şi de aceea s-a creat echipa de investigare care a dezvoltat mai multe tipuri de detectare automată, bazată pe semnături, comportament, comportament pe reţea, similaritate de cod(aceeaşi funcţionalitate refactorizat) şi modele de machine learning.

S-au creat și niște reguli mai severe care avertizează imediat utilizatorii că o aplicație a fost blocată și dezactivată.

25.
Actor super sofisticate
Cu prezență în industrie
Cu reputație
Un site web
Un proiect legitim, în ghilimele, dar prin care pot să arate că pot să facă funcționalități benigne/dorite
O echipă de ingineri cu procese de dezvoltare și release foarte inteligente
Nu fac o mulțime de greșeli
Nu urmăresc vreo țintă ușoară
Sunt foarte sistematici
Vin cu soluții tehnice foarte sofisticate. Etapa 3 din biblioteca de anti-analiză a fost cu siguranță una dintre cele mai sofisticate.
Infrastructură foarte matură
Capabili să creeze automat server de C2 cu configurațiile lor pe furnizori de servicii de cloud diferiți

26.
Dar sunt foarte Stealthy
Trafic de rețea criptat asimetric
Toate payload-urile malițioase sunt descărcate dinamic
Odată ce despachetează/dezarhivează o etapă o șterg imediat
Intr-un cuvânt smart.

27.
Unul dintre cele mai interesante lucruri care a fost, este cât de repede au răspuns la fiecare dintre diferitele modificări.
Am discutat despre cum iterează prin fiecare factor pentru a vedea ce detectează Antivirusul prin schimbarile din APK-ul din etapa 1, dar cel mai interesant este că după o prezentare a bibliotecii de anti-analiza, în 72 de ore au început să modifice toate IOC(Indicator de Compromis) -urile discutate în acea prezentare.

Totodată în etapele de anti-analysis au introdus semnături pentru mediul de analiza automat al Google care corespund unor setări și proprietăți.

Cu toate acestea Chamois este acum în stadiul de mentenanță și control.

29.
Maxim Martie 2018 cu 20.35 milioane de dispozitive infectate.
A scăzut până la 1.81 milioane în Martie 2019.

30.
Munca lor însă nu a încetinit și în aceeași perioadă de timp, au lansat 12800 de noi eșantione în ecosistem. Astfel in martie 2019 erau 27000 de eșantione de Chamois în ecosistem.
Fiecare eșantion fiind unic.

31.
Dar numărul de infectări a scăzut cu 91 % față de cele 20,35 milioane din Martie 2018 la

1.81 milioane in Martie 2019.

Din acest motiv Chamois este cel mai mare Botnet despre care nu ați auzit niciodată.

32.

33.
.