CENTER FOR CYBERSIKKERHED

UNDERSØGELSESRAPPORT

GLEMMER DU, SÅ HUSKER HACKERNE

Dansk organisation hacket fem gange på to år – tusindvis af danske servere stadig sårbare

Indhold

Resumé....3
Indledning....3
Dansk organisation hacket fem gange på to år....4
De fem kompromitteringer: Motiver og angrebsmønstre....5
Tusindvis af danske servere stadig sårbare....10
Anbefalinger: Forsvar i dybden....14
Bilag 1: Cyber Kill Chain®-modellen....22
Bilag 2: Tabel 1 (stor version) - Angrebsmønstre for de fem kompromitteringer....23
Bilag 3: Figur 3 (stor version) - Aktive sårbarheder i Danmark pr. 2. december 2019....24
Bilag 4: Figur 4 (stor version) - Sårbarheder i Danmark fordelt efter deres offentliggørelsestidspunkt....25

CENTER FOR CYBERSIKKERHED

Kastellet 30
2100 København Ø
Telefon: + 45 3332 5580
E-mail: cfcs@cfcs.dk

2. udgave maj 2022.

Formål
Denne undersøgelsesrapport beskriver en konkret sag, hvor en server hos en dansk organisation blev hacket fem gange på to år grundet manglende opdateringer af software med kendte sårbarheder.

Sagen er langt fra enestående, og rapporten giver forslag til tiltag, som kan hjælpe myndigheder og virksomheder til at beskytte sig bedre i fremtiden. Målgruppen for denne rapport er it-ledelse og it-teknikere.

Resumé

- En server hos en dansk organisation er blevet hacket fem gange mellem december 2016 og januar 2019.
- Hackerne brugte en kendt sårbarhed, som havde været mulig at lukke i mere end seks år.
- Én hacker forsøgte at stjæle computerkraft til kryptomining. Tre andre udnyttede ikke deres adgang, mens den sidste hacker forsøgte at trænge dybere ind i mere sensitive dele af netværket.
- Serveren blev imidlertid taget af internettet, før hackeren formåede at udvide sin adgang, og organisationens antivirusprogram forhindrede kryptomineren i at fungere.
- CFCS vurderer, det er mindre sandsynligt, at hackerne var statsstøttede.
- Ved hjælp af åbne kilder har CFCS identificeret tusindvis af lignende danske servere, som stadig har alvorlige sårbarheder, selvom opdateringer har været tilgængelige i adskillige år.

Indledning

Hackere har ofte held med at benytte ældre, kendte sårbarheder til at kompromittere danske myndigheder og virksomheder. I mange tilfælde har sikkerhedsopdateringer været tilgængelige i adskillige år forud for et angreb, men er aldrig blevet implementeret.

I denne rapport stiller vi derfor skarpt på udfordringen med ældre, kendte sårbarheder. CFCS ser ofte, at myndigheder og virksomheder har servere, der indeholder mindre sensitiv data, og derfor ikke omfattes af samme sikkerhedsforanstaltninger, som de mere kritiske dele af organisationens netværk. Selvom disse servere i sig selv kan virke som uinteressante mål, kan de dog fungere som et springbræt for hackere ind til mere sensitive dele af organisationen.

Rapporten beskriver en konkret sag, hvor en server hos en dansk organisation blev hacket fem gange på to år med en sårbarhed, som havde været mulig at lukke med opdateringer i mere end seks år.

Rapporten er opdelt i fire dele. Først beskrives sagen i korte træk, hvorefter hackernes angrebsmønster og motiver analyseres mere detaljeret. I tredje del illustreres, hvor let hackere kan identificere sårbare servere i Danmark ved hjælp af åbne og lettilgængelige søgemaskiner. Rapporten afsluttes med en række konkrete forslag til tiltag, som kan hjælpe myndigheder og virksomheder til at beskytte sig selv bedre i fremtiden.

Dansk organisation hacket fem gange på to år

I januar 2019 gik en alarm i CFCS' sensornetværk, som fangede et hackerangreb mod en dansk organisation. CFCS' undersøgelser viste, at serveren var blevet hacket fem gange på to år med første kompromittering tilbage i december 2016. Det er meget sandsynligt, at der var tale om fem uafhængige kompromitteringer, og det er mindre sandsynligt, at de var udført af statsstøttede hackergrupper.

Den kompromitterede server var en Windows 2003 server med en tilhørende Oracle Application Server installeret. Applikationen anvendes bl.a. til at generere en række interne statistiske rapporter. Alle fem hackere udnyttede en sårbarhed i denne applikation, der har været kendt siden 2012, som gav hackerne mulighed for at placere ondsindede filer direkte på serveren over internettet. Der er mere præcist tale om sårbarheden CVE-2012-3152.

Selvom sikkerhedsopdateringer til applikationen havde været tilgængelige i over seks år, blev applikationen imidlertid aldrig opdateret. Serveren havde således været sårbar mindst siden oktober 2012 - blot fem måneder efter at den blev opsat.

CFCS' sensornetværk

CFCS monitorerer løbende netværkstrafikken til og fra myndigheder og virksomheder, der er tilsluttet sensornetværket.

Indsatsen er målrettet særligt avancerede statsstøttede og andre aktører med betydelige ressourcer, og CFCS bistår i tilfælde af alvorlige hackerangreb.

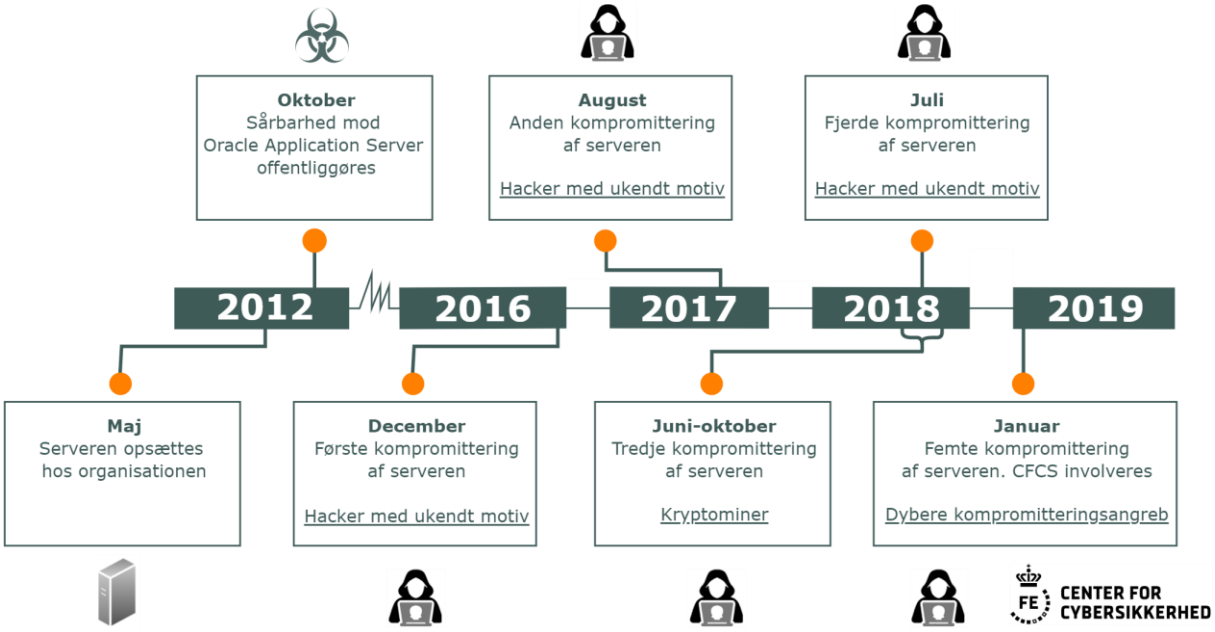
Der monitoreres kun i begrænset omfang for alment kendte angrebstyper, idet de pågældende organisationer selv forventes at have passende modforanstaltninger etableret.

Selvom data på serveren ikke i sig selv var af beskyttelsesværdig karakter, så forsøgte én hacker at få adgang til mere sensitive dele af netværket. En anden hacker forsøgte at stjæle computerkraft til at genere kryptovaluta, mens de tre sidste hackere ikke udnyttede deres ulovlige adgang til serveren.

På CFCS' anvisning blev serveren taget af internettet, før hackeren formåede at trænge dybere ind i netværket, og organisationens antivirusprogram stoppede kryptomineren i at fungere. Idet de sidste tre hackere aldrig udnyttede deres adgang, er det muligt, at der var tale om såkaldte "grey hat hackere", som blot ville teste, hvorvidt serveren kunne kompromitteres eller ej. Det er dog også muligt, at disse hackere kompromitterede serveren med henblik på at etablere en langsigtet adgang, som eventuelt kunne videresælges eller udnyttes i fremtiden.

Sårbarheden i Oracle applikationen var ikke serverens eneste sårbarhed, idet Microsoft allerede i 2015 officielt betragtede deres Windows 2003 server som forældet, og derefter stoppede med at levere sikkerhedsopdateringer til den.

Figur 1 giver et overblik over de væsentligste begivenheder i sagen, fra sårbarheden blev offentliggjort i 2012 og frem til den sidste kompromittering i januar 2019, hvor CFCS blev involveret, og serveren blev taget af internettet.



Figur 1: Tidslinje over væsentlige begivenheder

De fem kompromitteringer: Motiver og angrebsmønstre

I dette afsnit analyseres hackernes motiver og angrebsmønstre. Til analysen anvendes en analytisk model kaldet Cyber Kill Chain®-modellen, som oprindeligt blev udviklet af den amerikanske forsvars- og sikkerhedskoncern Lockheed Martin. Modellen opdeler et cyberangreb i en række kronologiske faser fra hackernes indledende rekognoscering af potentielle ofre til realisering af deres endelige mål med angrebet. Modellen har oprindeligt syv faser, men i denne rapport har vi valgt at slå afsendelses- og udnyttelsesfasen sammen for at lette formidlingen. Se bilag 1 for nærmere introduktion til modellen.

Selvom alle fem hackere indledningsvist brugte samme sårbarhed til at kompromittere serveren, så varierede deres adfærd betydeligt efter, at de havde opnået deres ulovlige adgang til serveren. Tabel 1 illustrerer således forskelle og ligheder i de fem kompromitteringer opdelt i de seks anvendte faser i Cyber Kill Chain®-modellen, som diskuteres enkeltvis nedenfor.

Fase	Hacker 1 December 2016	Hacker 2 August 2017	Hacker 3 Juni-oktober 2018	Hacker 4 Juli 2018	Hacker 5 Januar 2019
Rekognoscering	Identificerer sandsynligvis sårbare Oracle Application Servere via åben søgemaskine på internettet				
Klargøring	Bevæbner sig med ondsindet kode, der udnytter CVE-2012-3152 og webshells				
Afsendelse & Udnyttelse	Udnytter sårbarhed direkte fra egen server	Udnytter sårbarhed via kompromitteret server i Thailand	Udnytter sårbarhed via kompromitteret webserver i Mauritius	Udnytter sårbarhed direkte fra egen server	Udnytter sårbarhed via ekstern kompromitteret server
Installation	Downloader to standard webshells	Downloader én standard webshell	Downloader én standard webshell samt malware til kryptomining	Downloader én standard webshell	Downloader én China Chopper webshell samt malware til dybere rekognoscering
Kommando & Kontrol	Udnytter ikke webshells	Udnytter ikke webshell	Mange servere fra botnet tilgår automatisk webshell og tjekker status på kryptomineren	Udnytter ikke webshell	Tilgår webshell via ekstern kompromitteret server
Målorienterede Aktiviteter	Hacker med ukendt motiv	Hacker med ukendt motiv	Forsøger kryptomining, som dog forhindres af serverens antivirusprogram	Hacker med ukendt motiv	Foretager yderligere rekognoscering af netværk med henblik på dybere kompromittering

Tabel 1: Angrebsmønstre for de fem kompromitteringer

Rekognoscering

Det har ikke været muligt at fastlægge præcist, hvordan hackerne foretog deres indledende rekognoscering. Det er imidlertid sandsynligt, at hackerne fandt deres ofre ved hjælp af én af flere mulige åbne søgemaskiner, som enhver kan tilgå via internettet.

En af de mest udbredte søgemaskiner er Shodan, som løbende skanner internettet for opkoblede enheder, og indekserer dem i en åben database på deres hjemmeside. Shodan bruges bl.a. af sikkerhedseksperter til at finde sårbarheder i egne organisationer, men benyttes også ofte af hackere til at identificere sårbare mål.

Ved hjælp af Shodan eller en lignende service har hackerne dermed haft mulighed for at fremsøge internetvendte servere med en Oracle Application Server installeret, og således kunne generere en liste over potentielt sårbare mål.

Det er derfor sandsynligt, at organisationens server blev angrebet, blot fordi den fremgik i én af disse åbne databaser over potentielt sårbare servere. Kompromitteringerne var med andre ord sandsynligvis opportunistiske snarere end specifikt rettet mod organisationen.

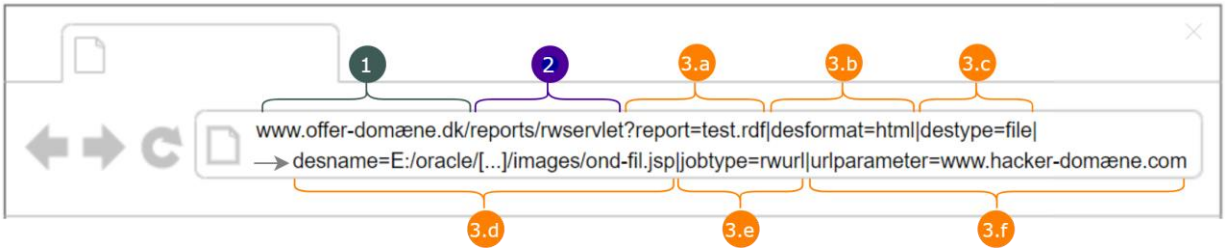
Klargøring

Efter at hackerne havde identificeret deres mål, klargjorde de den ondsindede kode, som kunne udnytte sårbarheden i Oracle Application Server-softwaren samt forskellige typer webshells.

Udfordringen for hackerne bestod i at placere webshells på serveren med henblik på at kunne fjernkontrollere den. Til denne udfordring anvendte alle hackerne sårbarheden i Oracle-applikationen kendt som CVE-2012-3152. Både den ondsindede kode og de anvendte webshells kunne frit downloades fra internettet.

Webshell
En webshell er et lille stykke programkode, der kan gemmes i en fil, og som giver fjernadgang via internettet.
Webshells fungerer derfor som et såkaldt "remote access tool" (RAT), der giver hackere mulighed for at læse, skrive, ændre, downloade eller slette filer på en server.

Sårbarheden CVE-2012-3152 omhandler mere specifikt et sikkerhedshul i rapportmodulet i Oracle-applikationen, som gav hackerne mulighed for at placere webshells direkte på organisationens server. I det følgende gennemgås en anonymiseret ikke-funktionel version af den ondsindede kode i korte træk:



Figur 2: Eksempel på ondsindet kode

Den ondsindede kode kan sendes direkte fra en almindelig webbrowser eller terminal, og kan med fordel diskuteres i tre delsektioner:

1. Først angav hackerne URL-adressen på den sårbare server for at kommunikere direkte til den.

2. Herefter specificerede hackerne, at de ønskede at rette deres kommunikation direkte til et konkret modul i Oracle-applikationen ved navn "Reports Servlet" forkortet "rwservlet" under hovedmodulet "reports". Modulet er beregnet til at generere forskellige statistiske rapporter, men hackerne havde fundet ud af, at det var muligt at placere vilkårlige filer i en specifik mappe beregnet til at opbevare billeder.

3. I den tredje del af den ondsindede kode specificerede hackerne en række parametre, som imiterede, at de ønskede at generere en almindelig rapport, men som i virkeligheden downloadede en ondsindet fil fra et domæne kontrolleret af hackerne:
   - 3.a I parameteret "report" foregav hackerne, at de ønskede at generere en almindelig rapport.
   - 3.b I "desformat" specificerede de hvilken formatering, rapporten skulle have. I ovenstående eksempel vist i html-format.

3.c I "destype" specificerede de hvilken type rapport, der skulle genereres. I ovenstående eksempel som en almindelig fil.

3.d Parametret "desname", var egentligt beregnet til at specificere, hvor på serveren den endelige rapport skulle gemmes, men heri lå applikationens anden store sårbarhed: Normalt har applikationer indbyggede sikkerhedsbegrænsninger, som gør, at filer ikke kan gemmes direkte på en server. Hackerne havde imidlertid fundet ud af, at en særlig mappe, beregnet til opbevaring af billeder til rapporterne, ikke havde denne sikkerhedsbegrænsning. Ved at angive netop denne mappe som destination, kunne de således omgå de normale sikkerhedsbegrænsninger, og få lov til at downloade vilkårlige filer ned på serveren. Samtidig troede applikationen blot, at der blev genereret en almindelig rapport. Alle hackerne kunne således frit downloade deres webshells via deres kontrollerede domæner i JavaServer Pages (jsp) format til billede-mappen, som herefter kunne eksekveres ved at besøge netop denne sti i en webbrowser. I ovenstående eksempel gemte hackeren en jsp fil i billede-mappen med navnet ond-fil.jsp.

3.e Det er særligt i dette parameter, "jobtype", at første del af sårbarheden i modulet opstod. Jobtype-parameteret giver nemlig mulighed for, at rapporter indarbejder elementer fra offentlige hjemmesider. Denne funktion kan være brugbar for organisationer, der ønsker at danne rapporter ud fra oplysninger fra internettet, men funktionen gav samtidig hackerne mulighed for at henvise til hjemmesider kontrolleret af hackerne selv.

3.f I "urlparameter" kunne hackerne således angive det specifikke hackerkontrollerede domæne, som rapporten skulle hente information fra. Tricket for hackerne bestod derfor i at gemme ondsindede filer på disse hackerkontrollerede domæner, som således blev downloadet, når applikationen besøgte det.

Hackerne kunne altså udnytte Oracle-applikationens legitime mulighed for at indarbejde information fra offentlige hjemmesider til at henvise til hjemmesider kontrolleret af hackerne. Herefter downloadede hackerne webshells til en særlig ubeskyttet mappe på organisationens server. Myndigheder og virksomheder med Oracle Application Servere kan med fordel gennemse denne særlige billede-mappe for mistænkelige filer, idet disse kan indikere en kompromittering.

CFCS er bekendt med tilfælde, hvor en statsstøttet hackergruppe har anvendt netop denne sårbarhed i deres angreb mod en anden stat.

Afsendelse & Udnyttelse

Efter at hackerne både havde identificeret sårbare ofre og klargjort koden, igangsatte de angrebet direkte mod organisationens server. Mere specifikt afsendte hacker 1 og 4 den ondsindede kode til serveren direkte fra deres egen maskine, mens hacker 2, 3 og 5 forsøgte at skjule deres identitet ved at gemme sig bag andre kompromitterede servere.

CFCS har sporet aktør 2 tilbage til en lignende Oracle server i Thailand, som tidligere er blevet kompromitteret, og som nu anvendes til at sende webshells ud til andre servere i verden.

Aktør 3 kontaktede derimod den sårbare statslige server via en webserver i Mauritius, som sandsynligvis var kompromitteret under en global kampagne kendt som Drupalgeddon 2.

Aktør 5 anvendte ligeledes en kompromitteret ekstern server for at sløre sin identitet.

Installation

Alle hackere udnyttede sårbarheden til at downloade webshells til serveren. Hermed lykkedes det dem at etablere en langsigtet ulovlig adgang til serveren, hvorfra de kunne kontrollere og uploade flere ondsindede filer.

De fire første hackere anvendte almindelige webshells, mens den sidste hacker downloadede en mere speciel type webshell kendt som en China Chopper. Det særlige ved en China Chopper webshell er bl.a. dens beskedne størrelse på den kompromitterede server sammenholdt med dens mange egenskaber. China Chopper webshellen har formentlig fået sit navn, da den oprindeligt særligt blev associeret med kinesiske cyberkriminelle, men brugen af China Chopper er efterhånden blevet mere udbredt.

Efter de havde lagt webshells på serveren, stoppede hacker 2 og 4 deres angreb, og hacker 1 udnyttede blot sin adgang til at uploade en ekstra webshell. Det er derfor muligt at disse tre hackere var såkaldte 'grey hat hackere', som blot ville se, om de kunne hacke serveren eller ej uden nødvendigvis at være drevet af et større formål med kompromitteringen. Det er dog også muligt, at hackerne kompromitterede serveren med henblik på at etablere en mere langsigtet adgang, som eventuelt kunne udnyttes i fremtiden eller videresælges.

Hacker 3 uploadede en lang række yderligere malware til serveren med henblik på at igangsætte en kryptominingsproces. Der blev bl.a. uploadet en kendt kryptominer af typen XMRig, som anvendes til at generere Bitcoins samt en fil, der løbende kunne eksekveres for at tjekke status på kryptomineren. Det er således meget sandsynligt, at motivet bag dette angreb var økonomisk.

Kryptomining
Kryptomining er en proces, hvor man sætter én eller flere computere til at løse meget komplicerede matematiske udregninger, som efterfølgende "belønnes" med kryptovaluta, fordi beregningerne hjælper med at understøtte valutaens blockchain. Kriminelle har således et økonomisk incitament til at stjæle computerkraft fra andre for at udføre disse udregninger for sig.

Hacker 5 placerede ligeledes flere forskellige typer malware på serveren, der bl.a. kunne anvendes til at foretage mere dybdegående rekognoscering af serverens indstillinger og netværk. Dette kunne indikere, at hackeren besluttede sig for at forsøge at trænge dybere ind i mere sensitive dele af netværket – muligvis fordi vedkommende opdagede, at serveren tilhørte en dansk organisation. Selvom denne hacker udviste mere avanceret adfærd relativt til de andre, er det dog stadig mindre sandsynligt, at der var tale om en statsstøttet hackergruppe.

Kommando & Kontrol

Hacker 1, 2 og 4 udnyttede ikke deres webshells efter, at de var blevet lagt på serveren.

Hacker 3 anvendte en mere kompleks Kommando & Kontrol opsætning, hvor en række kompromitterede servere i et kendt botnet løbende tilgik webshellen for at tjekke status på kryptomineren. Efter at hackeren havde etableret adgang til serveren med webshellen, overlod vedkommende således selve driften af kryptomineren til et automatiseret netværk af kompromitterede servere.

Hacker 5 tilgik sin China Chopper webshell gennem samme eksterne kompromitterede server, som blev anvendt til at sende den ondsindede kode.

Målorienterede Aktiviteter

Alle fem hackere formåede at kompromittere serveren, og opnåede direkte kontrol med den over internettet via deres webshells.

Hacker 1, 2 og 4 udnyttede som nævnt ikke deres adgang yderligere.

Hacker 3 forsøgte aktivt at udnytte serverens computerkraft til at generere Bitcoins for at opnå økonomisk gevinst, men antivirusprogrammet på serveren forhindrede kryptomineren i at fungere. Efter knap tre måneder stoppede botnettet med at tilgå serveren, sandsynligvis fordi hackeren havde opdaget, at kryptomineren ikke virkede. Derfor indstillede hackeren sit angreb.

Hacker 5 foretog målrettet rekognoscering af serveren og dens netværksindstillinger, sandsynligvis med henblik på at trænge dybere ind i mere sensitive dele af netværket. CFCS' varsel og efterfølgende frakobling af serverens adgang til internettet stoppede imidlertid hackeren, før det lykkedes vedkommende i at tilgå mere sensitive dele af netværket.

Tusindvis af danske servere stadig sårbare

Ovenstående sag er kun én blandt mange lignende sager, som CFCS løbende er i berøring med. Hackere udnytter meget ofte ældre sårbarheder i deres angreb, og i dette afsnit illustreres, hvor let det er at identificere disse sårbare enheder ved hjælp af åbne søgemaskiner såsom Shodan.

En søgning i Shodans database returnerer over 750.000 danske IP-adresser per december 2019 på alt fra webservere og webkameraer til industrielle kontrolanlæg, som alle har forbindelse til internettet. For hver IP-adresse oplistes en lang række informationer, der beskriver enheden bag IP-adressen, herunder bl.a.

hvilken hardware der er tale om, hvilken software, der er installeret på den, hvor enheden er lokaliseret, og i nogle tilfælde også hvem der ejer den.

Dét, der imidlertid gør Shodan og lignende services særligt interessante for hackere, er, at søgemaskinernes resultater aktivt sammenkobles med de sårbarheder, der gennem tiden er fundet på netop denne type enhed. Søgemaskinerne fortæller med andre ord ikke blot, hvordan man kommunikerer med forskellige internetopkoblede enheder, men også hvilke sårbarheder man formentlig kan udnytte for at hacke dem. Denne transparens er blevet et grundvilkår for både de myndigheder og virksomheder, som forsøger at beskytte sig, men også for de hackere, som ønsker at kompromittere dem.

De fem mest udbredte sårbarheder i Danmark

CVE-2018-1312 (14.525)
- Uhensigtsmæssig godkendelse

CVE-2017-7679 (12.975)
- Ubegrænset hukommelsesbuffer

CVE-2016-8612 (11.272)
- Uhensigtsmæssig inputvalidering.

CVE-2016-4975 (10.938)
- Mulig manipulering af linjeskift i kode

CVE-2019-0220 (10.287)
- Mulig manipulering af stiangivelser i URL'er

Shodan har imidlertid kun mulighed for at skanne en mindre del af det samlede antal danske IP-adresser. Det skyldes bl.a. at de fleste routere, som er den centrale indgang til et netværk, i dag ofte afviser indgående trafik, medmindre man selv har etableret en udadgående forbindelse først eller hvis enheden i kraft af sin funktion kræver indgående trafik, såsom f.eks. en webserver eller et webkamera med fjernadgang. Langt de fleste privatpersoner fremgår derfor heller ikke af disse databaser, og der er en overvægt af enheder, som på den ene eller anden måde kræver en åben og direkte kommunikationskanal.

Desuden skal Shodans indeksering af sårbare enheder tages med visse forbehold, idet den udelukkende baserer sig på enhedens software og versioner. Typisk vil sårbarheder omhandle specifikke funktioner i software, og hvorvidt enheden er sårbarbar eller ej afhænger således af, hvorvidt disse funktioner er slået til eller ej. Det siger Shodans skanning ikke noget om. Shodans angivelse af en sårbarhed betyder derfor ikke nødvendigvis, at enheden rent faktisk er sårbar, men snarere at enheden kan have denne sårbarhed. På trods af disse begrænsninger anvendes søgemaskinerne bredt af hackere, og Shodans data giver derfor et godt indblik i det landskab, som hackerne ofte orienterer sig i, når de foretager rekognoscering efter mulige ofre.

Ifølge Shodan var der i starten af december 2019 knap 1.600 forskellige aktive sårbarheder i Danmark fordelt på de omkring 750.000 indekserede danske enheder. Sårbarhederne varierer efter hvor mange enheder de hver især er udbredt på. Der er forskel på, hvor alvorlige de forskellige sårbarheder er, og hver enkelt enhed kan godt have mere end én sårbarhed på samme tid.

De fem mest alvorlige sårbarheder i Danmark med CVSS score på 10 rangeret efter udbredelse

CVE-2010-3972 (6.519)
- Buffer overflow

CVE-2010-0425 (1.262)
- Eksekvering af kode

CVE-2008-0075 (932)
- Eksekvering af kode

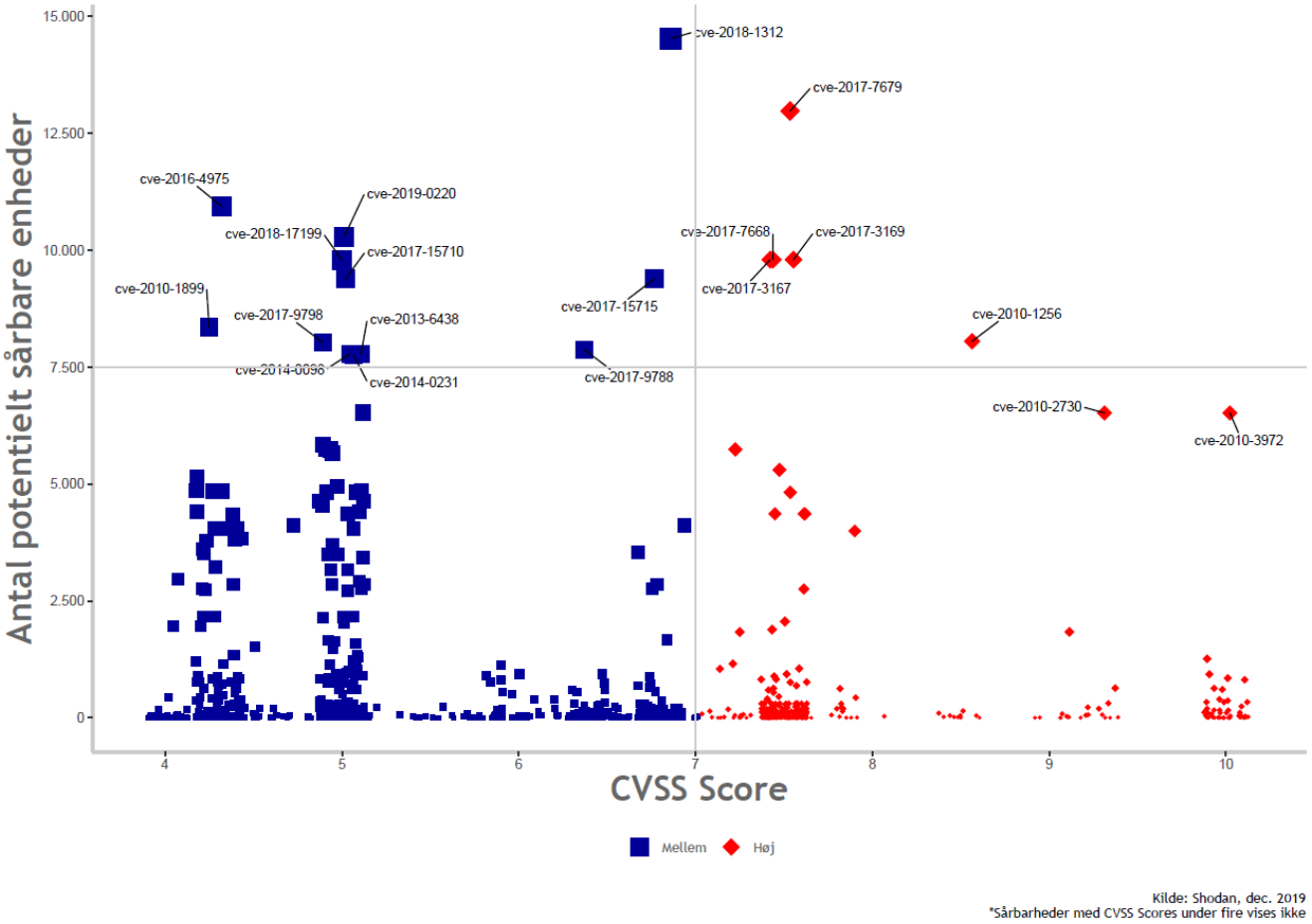
CVE-2012-2688 (848)
- Buffer overflow

CVE-2012-2376 (813)
- Buffer overflow

For at vurdere sårbarhedernes alvorlighed anvendes et scoringssystem udviklet af det amerikanske National Institute of Standards and Technology (NIST). NIST har to scoringssystemer, hvor denne rapport tager udgangspunkt i version 2.0, da hovedfokus her er på ældre sårbarheder. Udregningsmodellen sammenligner hver enkelt sårbarheds specifikke karakteristika, og rangerer dem på en skala fra 0-10, hvor værdier mellem 0,0-3,9 angiver sårbarheder med "lav" alvorlighed, og 4,0-6,9 er sårbarheder af "mellem" alvorlighed. Værdier fra 7,0-10,0 angiver sårbarheder med "høj" alvorlighed. En række faktorer indgår i udregningen af denne såkaldte "CVSS Score" herunder bl.a. hvilke rettigheder,

sårbarheden giver hackerne, hvor kompleks udnyttelsen af sårbarheden er i praksis, samt hvor bred adgang til den sårbare enheds information, hackerne kan få indsigt i. For yderligere information om udregningen af scoren se: https://nvd.nist[.]gov/vuln-metrics/cvss.

Figur 3 illustrerer de knap 1.600 aktive sårbarheder i Danmark med angivelse af alvorligheden for den enkelte sårbarhed.



Kilde: Shodan, dec. 2019
*Sårbarheder med CVSS Scores under fire vises ikke

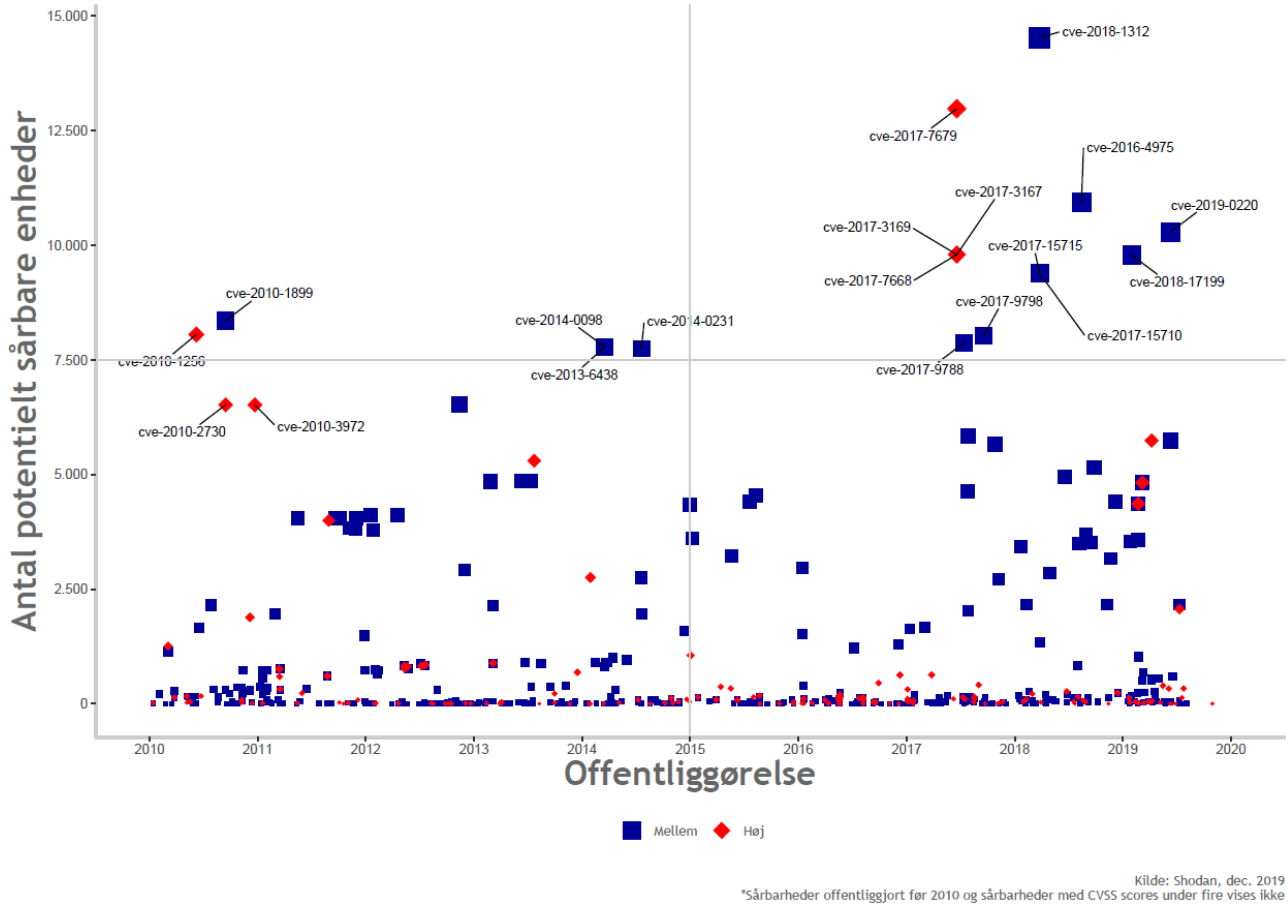
Figur 3: Aktive sårbarheder i Danmark pr. 2. december 2019

Baseret på Shodans data var de mest bekymrende sårbarheder i Danmark dem, som er placeret i figurens øverste højre hjørne, som både var udbredt på mere end 7.500 potentielt sårbare enheder, og som var af "høj" alvorlighed. I alt faldt fem sårbarheder inde for denne kvadrant. Desuden tangerede CVE-2010-2730 og CVE-2010-3972 til kvadranten. Disse to særligt alvorlige sårbarheder, fandtes på mere end 6.500 sårbare enheder i Danmark og havde CVSS Scores over ni.

Fra et samfundsmæssigt perspektiv er der naturligvis forskel på, hvorvidt en sårbar enhed understøtter kritisk infrastruktur eller tilhører mindre vigtige dele af samfundet. I den forstand kan en sårbarhed af "mellem" alvorlighed i kritisk infrastruktur godt få langt større samfundsmæssig effekt, hvis den bliver udnyttet, sammenlignet med en sårbarhed af "høj" alvorlighed i en mindre betydningsfuld enhed. Uagtet dette er det bemærkelsesværdigt, at der ifølge Shodan potentielt var flere tusinde enheder i Danmark med sårbarheder af "høj" alvorlighed, som muligvis ville kunne kompromitteres af hackere.

Antallet af sårbare enheder i Danmark afhænger i høj grad af vores evne til at implementere nødvendige sikkerhedsopdateringer. Tiden, der går fra offentliggørelsen af en sårbarhed, til sikkerhedsopdateringer bliver implementeret, er vigtig, fordi den bestemmer, hvor lang tid hackerne har til at udnytte sårbarhederne.

I sagen gennemgået i denne rapport udnyttede fem forskellige hackere samme sårbarhed, som var hele seks år gammel, og selvom Shodan ikke finder den konkrete sårbarhed på danske IP-adresser i dag, så illustrer figur 4, at sagen langt fra er enestående. Figur 4 viser de knap 1.600 forskellige aktive sårbarheder fordelt efter, hvornår de blev offentliggjort. Det giver os et billede af, hvor lang tid der går, før danske myndigheder og virksomheder implementerer vigtige sikkerhedsopdateringer, og dermed også hvor mange ældre sårbarheder, der stadig florerer i Danmark.



Kilde: Shodan, dec. 2019
*Sårbarheder offentliggjort før 2010 og sårbarheder med CVSS scores under fire vises ikke

Figur 4: Sårbarheder i Danmark fordelt efter deres offentliggørelsestidspunkt

Idet offentliggørelsen af sårbarheder ofte hurtigt følges af sikkerhedsopdateringer fra leverandøren, kunne man forvente et skarpt fald i antallet af aktive sårbarheder i løbet af de første par måneder. Grafen viser imidlertid en anden virkelighed, idet de fleste aktive sårbarheder i Danmark er adskillige år gamle. Enkelte strækker sig helt tilbage til 1999, og i mange tilfælde er de endda også relativt alvorlige sårbarheder.

Det betyder, at hackere ofte har flere år til at planlægge og udnytte ældre sårbarheder, før danske myndigheder og virksomheder opdaterer sig mod dem. I et bekymrende modsætningsforhold ser CFCS også hackere udnytte sårbarheder, som kun er få uger eller dage gamle.

I den afsluttende del af rapporten præsenteres en række initiativer, som kan hjælpe myndigheder og virksomheder til at beskytte sig selv bedre i fremtiden.

Citrix sårbarhed efterlod mere end tusind danske enheder potentielt sårbare

17. december 2019 blev en alvorlig sårbarhed i Citrix-netværksudstyr annonceret. Allerede 10. januar blev en vejledning i at hacke ved hjælp af sårbarheden - et såkaldt proof-of-concept – offentliggjort. Hackerne havde over en uge til at udnytte sårbarheden med hjælp fra vejledningen, før leverandøren udsendte første sikkerhedsopdatering 19. januar. CFCS er bekendt med flere danske organisationer, der i perioden er forsøgt kompromitteret.

En søgning i Shodans database ved offentliggørelsen af sårbarheden viste over tusind potentielt sårbare enheder i Danmark.

CFCS har udsendt varsler til identificerede sårbare myndigheder og virksomheder, og løbende vejledt om modforanstaltninger.

Anbefalinger: Forsvar i dybden

Ved at anvende Cyber Kill Chain®-modellen i analysen er det muligt at lære af ovenstående sag og forbedre sin cybersikkerhed på struktureret vis. Mere specifikt kan opdelingen af angrebet i enkeltstående faser identificere nogle af de sikkerhedsforanstaltninger, som kunne have stoppet eller hæmmet angrebet i hver fase af angrebet.

I det følgende præsenteres en række tiltag, som myndigheder og virksomheder med fordel kan implementere for at beskytte sig bedre mod lignende angreb. Initiativerne er funderet i konceptet "forsvar i dybden", som betoner, at det aldrig vil være muligt at gardere sig fuldstændigt fra et målrettet hackerangreb, men at det er muligt at øge sin sandsynlighed for at opdage, forhindre eller begrænse det ved at ved at etablere forskellige lag af sikkerhed i sine it-systemer. Tabel 2 giver et overblik over de forskellige initiativer i hver angrebsfase, som præsenteres enkeltvist i det følgende.

Fase	Opdag	Imødegå
Rekognoscering	• Brug åbne databaser i eget cyberforsvar	• Overvej afskærmning af udstyr • Begræns adgang og bannerinformation for søgemaskiner
Klargøring	• Hold dig informeret om nye sårbarheder	
Afsendelse & Udnyttelse	• Network/Host Intrusion Detection Systems - NIDS & HIDS • File integrity checking	• Anvend ikke standard login/password • Patching: Hold dine systemer opdaterede • Network/Host Intrusion Prevention Systems - NIPS & HIPS • Firewalls
Installation		• Antivirus • Adgangskontrol
Kommando & Kontrol		• Forhindre mistænkelig ind- og udgående trafik
Målorienterede Aktiviteter	• Logning	

Tabel 2: Cybersikkerhedsinitiativer fordelt på angrebsfaser

Rekognoscering

Hackerne identificerede som nævnt sandsynligvis organisationens server ved hjælp af offentligt tilgængelige søgemaskiner som eksempelvis Shodan.

Opdag

Brug åbne søgemaskiner i eget cyberforsvar

Virksomheder og myndigheder kan med fordel benytte samme søgemaskiner, som hackerne anvender til at finde egne sårbare internetvendte enheder. Mange særligt større organisationers netværk har udviklet sin it-infrastruktur gennem adskillige år, og det kan være vanskeligt at bevare det fulde overblik. Disse søgemaskiner kan derfor være gode redskaber til at komme hackerne i forkøbet, og opdage "glemt" eller på anden vis sårbart udstyr på eget netværk, før hackerne finder det. Der findes flere af disse, men to af de mere udbredte findes på hjemmesiderne Shodan[.]io og Censys[.]io. En oplagt mulighed kan også være at hyre et it-sikkerhedsfirma til at scanne sin organisations systemer efter sårbart udstyr.

Såfremt man i sine søgninger finder eget sårbart udstyr i eksempelvis Shodan, og er i tvivl om hvordan man skal forholde sig, er man velkommen til at kontakte CFCS' situationscenter for vejledning per mail eller telefon på CFCS@CFCS.dk eller +45 33 32 55 80.

Imødegå

Overvej afskærmning af udstyr

Ikke alle enheder behøver forbindelse til og fra internettet for at kunne udfylde deres funktion. I nogle tilfælde er det nok, at enheden kan benyttes på organisationens interne lokale netværk (LAN) uden internetforbindelse. Jo flere internetopkoblede enheder, desto flere angrebsvinkler har hackere til at kompromittere organisationen.

Det er især hensigtsmæssigt at afskærme netværksudstyr fra internettet, hvis opdatering af udstyret ikke er mulig. Der kan være flere grunde til, at en opdatering ikke er mulig at gennemføre, og i disse tilfælde er det bedste alternativ, at enheden ikke kan tilgås fra internettet. Problematikken omkring systemer som ikke længere kan opdateres, også kaldet 'legacy systemer', er beskrevet yderligere nedenfor under afsnittet "Patching: Hold dine systemer opdaterede".

Begræns adgang og bannerinformation for søgemaskiner

Myndigheder og virksomheder kan med fordel begrænse den bannerinformation, som disse søgemaskiner opnår, når de foretager deres skanninger.

Alternativt kan myndigheder og virksomheder forsøge at nægte søgemaskinernes mulighed for at skanne sine enheder. IP-adresserne på skanningssidernes enheder kan eksempelvis tilføjes i organisationens firewall og derved blokere skanningen. Udfordringen er, at disse virksomheder forsøger at holde sine IP-adresser hemmelige og jævnligt udskifter dem netop for at undgå at blive blokeret. Listen af blokerede IP-adresser skal således løbende opdateres, hvilket kan være en krævende opgave, hvor indsatsen ikke nødvendigvis står mål med udbyttet. Til hjælp til at kortlægge skanningsmaskinerne henvises til hjemmesiden greynoise[.]io.

Klargøring

Hackerne klargjorde den ondsindede kode, som udnyttede sårbarheden i Oracle Application Server softwaren samt forskellige typer webshells, der begge kunne downloades frit fra internettet.

Opdag

Hold dig informeret om nye sårbarheder

Mange hackere opdager sårbarheder i online fora eller på hjemmesider, hvor de offentliggøres bl.a. som led i distribueringen af opdateringer. Det kan inspirere hackere til at afprøve disse i virkeligheden. Organisationer kan således med fordel selv følge med i sådanne fora på internettet, hvor de nyeste sårbarheder relateret til egne systemer offentliggøres, for på den måde at følge med i hackernes foretrukne angrebsmetoder. Man bør i den forbindelse være særligt opmærksom i de tilfælde, hvor sårbarheder offentliggøres med tilhørende Proof of Concept (PoC) kode, dvs. hvor den ondsindede kode stilles frit tilgængelig med tilhørende klare vejledninger i, hvordan den præcis anvendes i praksis.

Afsendelse & Udnyttelse

Hackerne afsendte den ondsindede kode for at udnytte sårbarheden i Oracle-applikationen hos organisationens server – nogle angreb direkte, mens andre benyttede et kompromitteret mellemled for at sløre deres identitet.

Opdag

Network og Host Intrusion Detection Systems – NIDS og HIDS

Intrusion Detection Systemer (IDS) er den samlede betegnelse for to forskellige løsninger, der overvåger aktivitet på enten netværks- (NIDS) eller host-niveau (HIDS), og som giver alarmer, hvis systemet får mistanke om ondsindet aktivitet. Når en alarm går, skal den manuelt analyseres for at afgøre, hvorvidt der er tale om reel ondsindet aktivitet eller blot en falsk positiv alarm.

Network Intrusion Detection Systemer (NIDS) virker på netværksniveau, hvor de monitorerer og analyserer netværkstrafikken for hele organisationens netværk. Her kigger de efter mønstre, som kan afsløre et angreb. Det er i den forbindelse vigtigt, at man har et godt kendskab til sin normale netværkstrafik for derved at kunne opdage, når noget uregelmæssigt finder sted. Det er desuden vigtigt, at man i sin organisation gør sig klart, hvad man gør i tilfælde af en alarm, herunder hvorvidt man har ressourcerne til at analysere de alarmer, som systemerne genererer eller ej. En alarm uden handling har ingen effekt.

Snort er en af de mest anvendte applikationer, som er designet til at understøtte identificeringen af mistænkelig netværkstrafik. Applikationen giver mere præcist mulighed for at opstille forskellige regler, der giver en advarsel, såfremt netværkstrafikken udviser særlige karakteristika. Kunsten er at udforme disse Snort-regler således, at de klart skelner mellem legitim og ondsindet netværkstrafik, og hermed begrænser omfanget af falsk positive alarmer, som skal analyseres nærmere. Man kan både skrive sine egne regler eller bruge regler, som andre har skrevet. Disse regler deles frit i det store online community, der eksisterer omkring Snort. Flere Snort-regler er udviklet netop med henblik på at spotte den trafik, der genereres af ondsindede webshells.

Host Intrusion Detection Systemer (HIDS) fungerer i princippet på samme måde som NIDS, men i stedet for at opdage angrebet i netværkstrafikken, opererer systemet på host- eller klientniveau - eksempelvis en PC - og dermed tættere på brugerne. Det betyder, at reglerne kan være mere specifikke, og det er f.eks. muligt her at opdage bagdøre på den enkelte host, som eksempelvis er blevet installeret gennem lokale WIFI- eller Bluetooth-forbindelser. Desuden kan angreb, der igangsættes indefra, eksempelvis ved brug af USB-stik, ligeledes opdages. Det kan dog være en stor opgave at implementere og understøtte disse systemer på host-niveau særligt i store komplekse organisationer.

Såkaldte YARA-regler kan ligeledes indbygges i ens HIDS, som er designet specifikt til at lede efter malwarekode på host-niveau. Når man kender en bestemt malwaretype og den underliggende kode, er det med andre ord muligt at skrive YARA-regler, der identificerer netop denne malware i fremtiden.

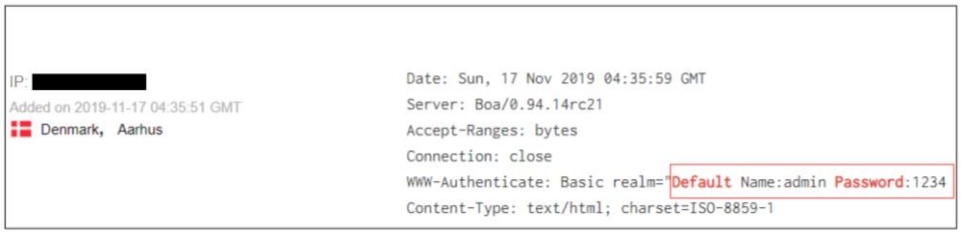
File integrity checking

File integrity checking betyder, at man definerer en liste over særligt kritiske filer, som man overvåger for ændringer. Det sker konkret ved, at man løbende genererer en kryptografisk hash-værdi for den enkelte fil, som derefter gemmes på et sikkert sted. Hvis én af de overvågede filer på et tidspunkt ændres, vil dette resultere i en ny hashværdi. På den måde kan man overvåge, hvorvidt der foretages ændringer i kritiske filer.

Imødegå

Anvend ikke standard login/password

Når man opsætter nye it-systemer, er det helt essentielt, at man skifter de standard-loginoplysninger, som hardwaren eller softwaren har fået tildelt af producenten. Det er meget nemt for hackere at finde disse login-oplysninger, som forskellige it-produkter "fødes med", da de som regel er frit tilgængelige på internettet. Figur 5 illustrerer et eksempel på, hvordan hackere deler login-information om danske servere frit på Shodans hjemmeside – i dette eksempel en server lokaliseret i Aarhus med brugernavnet "admin" og kodeordet "1234". Hvis standard-loginoplysningerne ikke udskiftes, er der ingen beskyttelse af den pågældende enhed mod kompromittering. Se CFCS' "Passwordvejledning" på vores hjemmeside for nærmere uddybelse.



Figur 5: Deling af loginoplysninger om danske servere via Shodan

Patching: Hold dine systemer opdaterede

At holde sine systemer opdaterede er et af de vigtigste elementer i et effektivt cyberforsvar. Hvis organisationens server i den konkrete sag var blevet opdateret med tilgængelige sikkerhedsopdateringer, havde hackerne ikke haft mulighed for at kompromittere serveren på den måde, som er beskrevet i denne rapport. Etableringen af et effektivt opdateringsregime kan dog være en udfordring af flere grunde.

For det første er kortlægningen af egen digital infrastruktur kompleks. De fleste organisationer anvender et omfattende antal it-systemer i deres daglige arbejde, og udviklingen af ny teknologi bevæger sig meget hurtigt. Man bør dog huske, at alle it-systemer og internetvendte enheder er en potentiel angrebsvinkel for hackere. At holde overblik over sit eget dynamiske it-landskab, og samtidig holde alle disse systemer opdaterede, kan være en udfordrende opgave, men er ikke desto mindre meget vigtigt for cybersikkerheden. Her kan man overveje at implementere et it-managementsystem, der kontinuerligt hjælper med at kortlægge hele organisationens digitale infrastruktur.

En anden udfordring forbundet med patching er en uklar ansvarsfordeling mellem organisationen og leverandører. Er det organisationen selv eller it-leverandørens ansvar at implementere og informere om kritiske sikkerhedsopdateringer? Ansvaret er ikke altid veldefineret, og resultatet bliver derfor, at visse it-systemer bliver glemt i gråzonen. Ansvaret for opdateringer af operativsystemer,

programmer og tredjepartskomponenter bør derfor altid være klart angivet i eksempelvis en patch-managementpolitik eller som kontraktligt definerede krav til leverandører. Se "Anbefalinger til styrkelse af sikkerheden i statens outsourcede it-drift" på CFCS' hjemmeside for nærmere uddybelse.

En tredje udfordring med patching er brugen af såkaldte 'legacy-systemer'. Nogle organisationer er afhængige af ældre, typisk specialudviklede it-systemer, som understøtter kritiske dele af organisationen, men som enten ikke længere vedligeholdes af leverandøren eller på anden vis kræver betydelige ressourcer at holde opdaterede. Disse 'legacy-systemer' udgør imidlertid en betydelig angrebsvektor af samme grund. Hvis det ikke er muligt at opdatere eller helt afskaffe sine 'legacy-systemer', bør man afskærme dem bedst muligt fra internettet eksempelvis ved at holde dem i et separat netværkssegment. Der bør derfor tages en informeret beslutning på ledelsesniveau om, hvorvidt organisationen kan leve med den sikkerhedsrisiko et upatchet system på netværket udgør for organisationen.

En fjerde udfordring ved patching er truslen fra software supply chain-angreb. Selvom hurtig implementering af sikkerhedsopdateringer er vigtig, bør man samtidig være opmærksom på muligheden for et software supply chain-angreb. I et software supply chain-angreb kompromitteres organisationer ved, at en angriber har kompromitteret selve it-leverandøren og herefter eksempelvis har placeret malware i en ny opdatering, som rulles ud til leverandørens kunder. Derfor bør man som udgangspunkt kun benytte anderkendte it-leverandører, som har et højt sikkerhedsniveau. Nogle leverandører offentliggør desuden hash-værdier i forbindelse med deres sikkerhedsopdateringer. Leverandørens kunder kan med disse krydstjekke, hvorvidt der er foretaget uautoriserede ændringer i opdateringen undervejs til kunden, som diskuteret ovenfor under afsnittet "File Integrity Checking".

Network og Host Intrusion Prevention Systems – NIPS og HIPS

Intrustion Prevention Systemer (IPS) er den samlede betegnelse for to forskellige sikkerhedsløsninger, som ikke blot overvåger netværksaktivitet, men også aktivt stopper mistænkelig trafik, der matcher de regler, som angives i systemet. Ulempen er, at man skal være endnu mere opmærksom på falsk positive hits, da disse vil blokere legitim trafik i netværket eller på host-niveau. Disse løsninger findes ligeledes både på netværks- (NIPS) eller host-niveau (HIPS), og fungerer efter samme principper som Intrusion Detecton Systermer diskuteret ovenfor.

Firewalls

Firewalls er en anden helt basal del af en organisations cybersikkerhed. Firewalls giver bl.a. mulighed for at åbne og lukke specifikke porte og dermed begrænse den internettrafik, som kommer både inde- og udefra organisationen. Firewalls kan desuden også bruges til at blokere specifikke domæner eller IP-adresser, som er kendt som værende ondsindede. En firewall er typisk placeret mellem internettet og organisationens lokale netværk, men kan også placeres andre steder. Firewalls kan eksempelvis også installeres på den enkelte host og styre hvilken trafik, der er tilladt til og fra hosten.

Firewalls bliver til tider anset som den vigtigste, og nogle gange eneste, sikkerhedsløsning, som man reelt behøver at have, men det er vigtigt at være bevidst om firewallens begrænsninger. Hackere kender ofte til den type trafik, der tillades gennem firewalls, og kan gemme sig i den trafik. Det er derfor vigtigt, at en

organisation også anvender andre metoder til at opdage og forhindre angreb. En firewall bør derfor aldrig stå alene, som blot eneste sikkerhedslag.

Installation

Hackerne downloadede både webshells og i nogle tilfælde også yderligere malware til serveren.

Imødegå

Antivirus

Et opdateret antivirusprogram er et vigtigt led for at opdage og stoppe trusler på host-niveau. En meget stor andel af den trafik – også den ondsindede - der sendes over nettet til servere er krypteret, og kan derfor være vanskelig at opdage for detekteringssystemer. Derfor er det vigtigt også at have et godt antivirus-system på host-niveau, som kan spotte ondsindet kode eller programmer, der kører direkte på maskinen, efter indholdet er blevet dekrypteret ved ankomst på serveren. I ovenstående eksempel afgav serverens antivirusprogram rent faktisk en del alarmer relateret til kryptomineren, men der blev ikke reageret på alarmerne, hvilket selvfølgelig var problematisk. Heldigvis fungerede organisationens specifikke antivirusprogram også som et HIPS, idet programmet aktivt stoppede kryptomineren i at blive kørt på serveren.

Antivirus er imidlertid heller ikke et tilstrækkeligt forsvar alene, da f.eks. webshells kan være vanskelige for antivirusprogrammer at opdage. IDS og IPS, herunder implementeringen af f.eks. YARA- og Snort-regler, kan derfor være et værdifuldt supplement til antivirusløsninger, som diskuteret ovenfor.

Adgangskontrol

Det er vigtigt at begrænse antallet af brugere med skriveadgang til serveren. Adgangslister specificerer hvilke brugere, der har hvilke rettigheder på organisationens forskellige systemer. Det er eksempelvis vigtigt at skelne mellem almindelige brugere og brugere med administratorrettigheder. Se mere om brugen af adgangskontrol i CFCS' vejledning "Cyberforsvar der virker" på vores hjemmeside.

Kommando & Kontrol

For at kontrollere serveren skulle hackerne tilgå deres webshells, og således oprette en Kommando & Kontrol forbindelse.

Imødegå

Forhindre mistænkelig ind- og udgående trafik

Trafik til disse kommunikationsforbindelser kan være svære at opdage, men nogle IDS'er kan opdage mistænkelig aktivitet, herunder suspekte DNS-opslag eller åbning af nye porte. Alternativt kan virksomheder og organisationer indføre såkaldt "white-" og "black-listing", hvor en netværksadministrator markerer

specifikke hjemmesider og applikationer, som værende enten betroede eller ondsindede og hermed kan vedkommende blokere eller tillade adgang til disse for organisationens almindelige brugere.

Målorienterede Aktiviteter

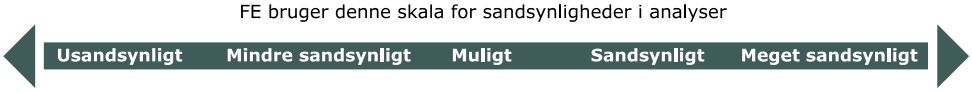
Opdag

Logning

Effektiv logning er en helt central del af et effektivt cyberforsvar. Logning er i princippet vigtig i alle faser af et angreb, men er placeret i denne fase for at illustrere, at det kun er igennem god logning, at man kan opdage og analysere et angreb, såfremt at hackeren er nået igennem alle de første faser.

Selvom man som organisation har implementeret mange af de ovenstående tiltag, kan man stadig ikke sikre sig fuldstændigt overfor cyberangreb. Når man er blevet udsat for et angreb, er det derfor helt afgørende, at man har gode logs, der kan hjælpe med at analysere, hvad der præcis er sket. Der kan være flere udfordringer i forbindelse med logning, såsom lagerkapacitet til at gemme de omfattende datamængder, og sikre at man logger den rigtige data. Desuden er organisationer underlagt forskellig lovgivning, der regulerer, hvilken data samt hvor lang tid data må opbevares. Disse udfordringer er beskrevet mere fyldestgørende i CFCS' vejledning "Logning – en del af et godt cyberforsvar" på vores hjemmeside.

FE bruger denne skala for sandsynligheder i analyser



Bilag 1: Cyber Kill Chain®-modellen

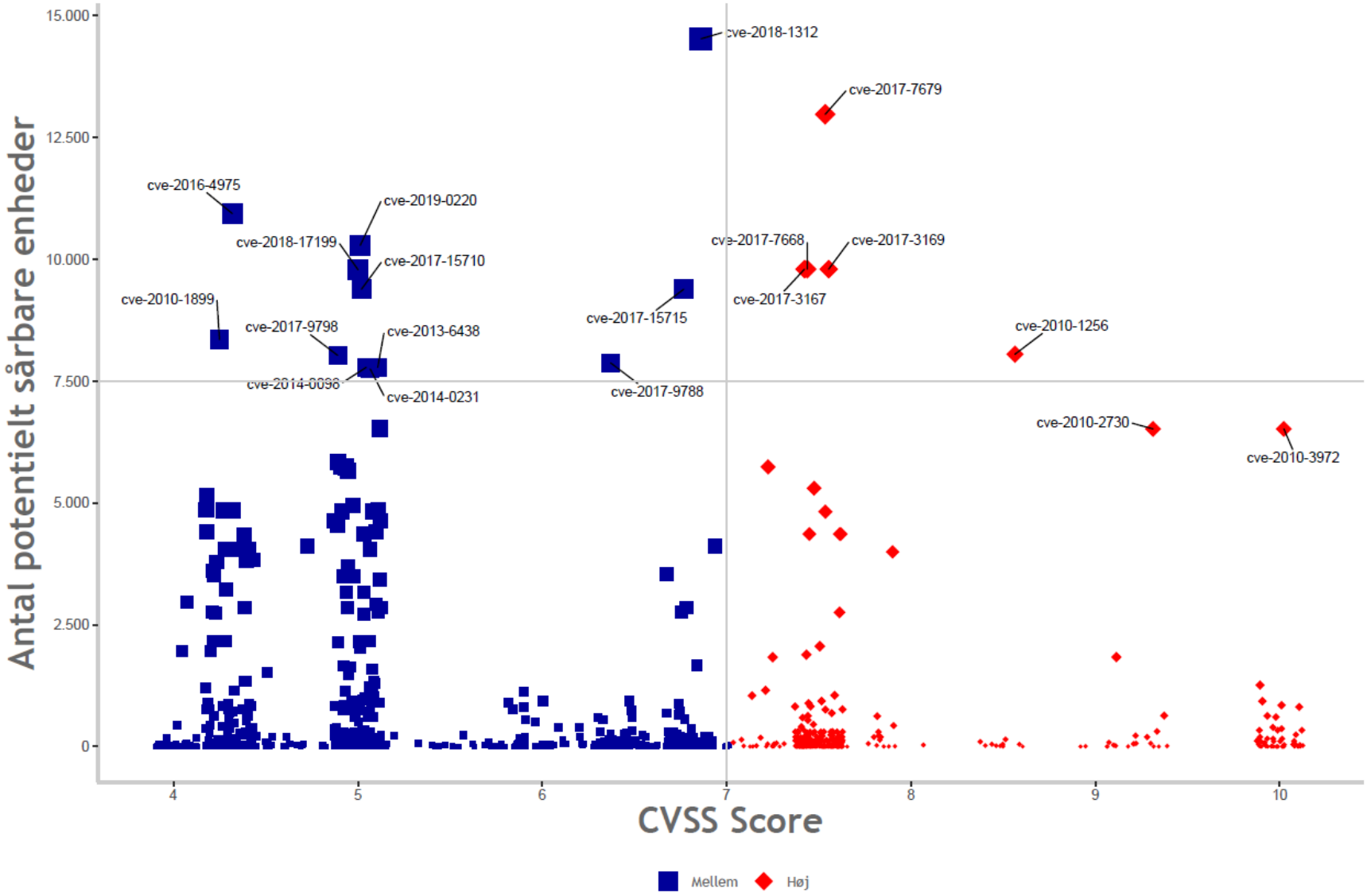
Cyber Kill Chain®-modellen giver en ramme til at analysere og forebygge cyberangreb. Modellen er oprindeligt udviklet af den amerikanske forsvars- og sikkerhedskoncern Lockheed Martin. Modellen anvendes bredt i både den offentlige og private sektor, og fastlægger oprindeligt syv kronologiske faser, som en hacker typisk gennemgår under et cyberangreb. I denne rapport er faserne "afsendelse" og "udnyttelse" imidlertid lagt sammen til én fase for at lette formidlingen. Ved at operationalisere en hackers aktiviteter som én sammenhængende 'kæde' af nødvendige aktiviteter, er det muligt at kortlægge angriberens specifikke angrebsmønster og udvikle skræddersyede sikkerhedsforanstaltninger til hver fase. De syv faser er:

- **Rekognoscering:** Hackeren planlægger sit angreb, og foretager den nødvendige research og identificerer potentielle ofre, som giver hackeren mulighed for at indfri sine mål. Ofre kan være alt fra en tilfældig sårbar server til målrettede angreb mod enkeltindivider eller organisationer af særlig interesse.
- **Klargøring:** På baggrund af den indledende rekognoscering gør hackeren sig klar til angreb ved at identificere eller selv udvikle de ondsindede redskaber, som kan bruges til at kompromittere sit offer.
- **Afsendelse:** Hackeren afsender ondsindede redskaber til sit offer f.eks. via e-mail, kompromitterede hjemmesider eller gennem USB-adgange.
- **Udnyttelse:** Redskaberne udnytter identificerede sårbarheder hos offeret.
- **Installation:** Når en hacker har tilegnet sig adgang, vil vedkommende typisk installere yderligere redskaber for at sikre sig en mere langsigtet adgang samt placere andre typer redskaber, som kan indfri mere specifikke mål.
- **Kommando & Kontrol:** De forskellige redskaber placeret hos offeret vil ofte etablere en kommunikationskanal, som angriberen anvender til at monitorere og styre sine operationer fra.
- **Målorienterede Aktiviteter:** Først efter at have gennemgået de seks indledende faser kan angriberen agere på sine oprindelige målsætninger. Disse varierer bredt fra spionage eller berigelseskriminalitet til direkte ødelæggelse af hardware.

Bilag 2: Tabel 1 (stor version) - Angrebsmønstre for de fem kompromitteringer

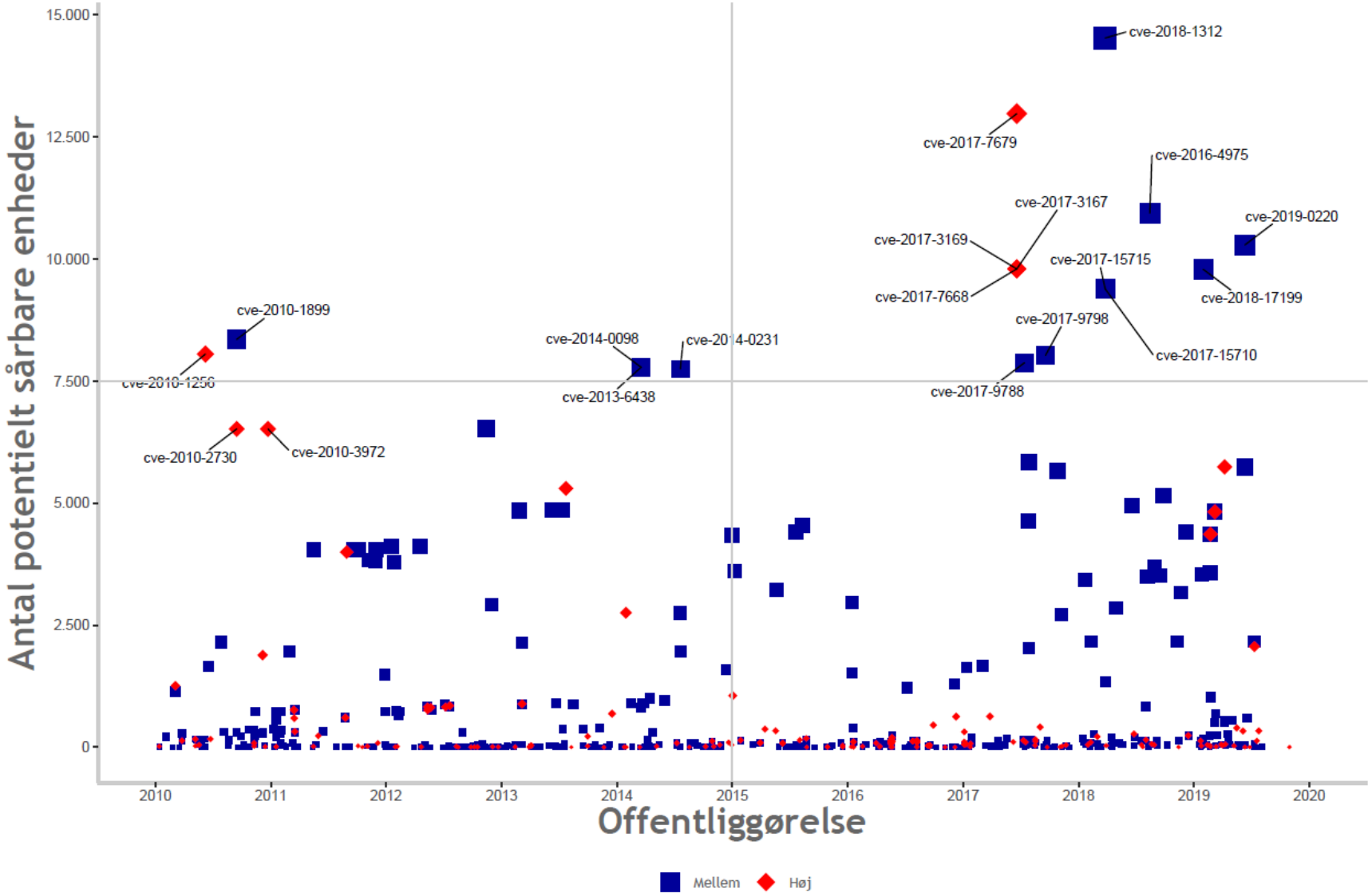
Fase	Hacker 1 December 2016	Hacker 2 August 2017	Hacker 3 Juni-oktober 2018	Hacker 4 Juli 2018	Hacker 5 Januar 2019
Rekognoscering	Identificerer sandsynligvis sårbare Oracle Application Servere via åben søgemaskine på internettet				
Klargøring	Bevæbner sig med ondsindet kode, der udnytter CVE-2012-3152 og webshells				
Afsendelse & Udnyttelse	Udnytter sårbarhed direkte fra egen server	Udnytter sårbarhed via kompromitteret server i Thailand	Udnytter sårbarhed via kompromitteret webserver i Mauritius	Udnytter sårbarhed direkte fra egen server	Udnytter sårbarhed via ekstern kompromitteret server
Installation	Downloader to standard webshells	Downloader én standard webshell	Downloader én standard webshell samt malware til kryptomining	Downloader én standard webshell	Downloader én China Chopper webshell samt malware til dybere rekognoscering
Kommando & Kontrol	Udnytter ikke webshells	Udnytter ikke webshell	Mange servere fra botnet tilgår automatisk webshell og tjekker status på kryptomineren	Udnytter ikke webshell	Tilgår webshell via ekstern kompromitteret server
Målorienterede Aktiviteter	Hacker med ukendt motiv	Hacker med ukendt motiv	Forsøger kryptomining, som dog forhindres af serverens antivirusprogram	Hacker med ukendt motiv	Foretager yderligere rekognoscering af netværk med henblik på dybere kompromittering

Bilag 3: Figur 3 (stor version) - Aktive sårbarheder i Danmark pr. 2. december 2019



Kilde: Shodan, dec. 2019
*Sårbarheder med CVSS Scores under fire vises ikke

Bilag 4: Figur 4 (stor version) - Sårbarheder i Danmark fordelt efter deres offentliggørelsestidspunkt



Kilde: Shodan, dec. 2019
*Sårbarheder offentliggjort før 2010 og sårbarheder med CVSS scores under fire vises ikke