مجازی سازی امن برای محاسبات ابری

چکیده

تصویب و انتشار محاسبات ابری جزء موارد امنیتی حل نشده است که ارائه‌دهنده و کاربران ابر را تحت تاثیر قرار می‌دهد. در این مقاله، نشان می‌دهیم که چگونه مجازی‌سازی می‌تواند با حفاظت از یکپارچگی ماشین‌های مجازی مهمان و اجزای زیرساخت ابری باعث افزایش امنیت محاسبات ابری شود. به طور خاص، یک معماری جدید، سیستم پیشرفته‌ی حفاظت از ابر(ACPS)، را پیشنهاد می‌کنیم که باعث تضمین افزایش امنیت در منابع ابر می‌شود. ACPS می‌تواند به‌طور موثر یکپارچگی مهمان و زیرساخت قطعات را نظارت کند، در حالی که هنوز به‌طورکامل به ماشین‌های مجازی و به کاربران ابر شفاف نیست. ACPS به صورت محلی می‌تواند به نقض امنیت به عنوان یک لایه بیشتر در مدیریت امنیت از چنین رویدادهایی واکنش نشان دهد. یک نمونه اولیه از پیشنهاد ما این است که ACPS به طور کامل در دو راه‌حل متن باز در حال اجرا است: Eucalyptus و OpenECP. نمونه اولیه در برابر اثربخشی و عملکرد تست شده است. به‌طور خاص: (الف) اثر نمونه‌ی آزمایش ما در برابر حملات شناخته شده نشان داده شده است؛ (ب) ارزیابی عملکرد نمونه اولیه ACPS تحت انواع مختلف از حجم کار انجام شده است. نتایج نشان می‌دهد که پیشنهاد ما در برابر حملات انعطاف پذیر است و سربار در مقایسه با امکانات ارائه شده کوچک است.

کلید واژه ها: امنیت، پردازش ابری، فناوری های مجازی سازی

1. معرفی

اینترنت در کنار انقلاب‌های دیگراست، که در آن منابع در سطح جهان شبکه شده است و به‌راحتی می‌توان به اشتراک گذاشت. پردازش ابری جزء اصلی از این دیاگرام است، که یک مخزن بزرگ از اینترنت را ارائه می‌کند که در آن منابع برای هر کسی به عنوان سرویس در دسترس است. به‌طور خاص، گره‌های ابر به طور فزاینده‌ای محبوب هستند اگرچه امنیت و حریم خصوصی مسائل حل نشده موجب کم شدن سرعت پذیرش و موفقیت آن شده است. در واقع، صداقت، محرمانه بودن و نگرانی در دسترس بودن، هنوز هم از مشکلات باز است که نیاز به پاسخ و راه‌حل‌های کارآمد و موثر می‌باشد. گره ابر ذاتا به حملات اینترنتی نسبت به راه‌حل‌های سنتی، اندازه‌ی آنها و خدمات مربوط به پیچیدگی که به ارمغان می‌آورد، آسیب‌پذیرتر است. در واقع، ابر اینترنتی با تمام جوانب مثبت و منفی یک سیستم فراگیر است. در نتیجه، افزایش حفاظت گره‌های ابر یک کار چالش برانگیز است. بنابراین به رسمیت شناختن تهدیدات برای ایجاد فرآیندهای امنیتی برای محافظت از خدمات و سیستم‌عامل‌های میزبان حملات بسیار مهم است.

محاسبات ابری در حال حاضر از اهرم مجازی‌سازی برای بار تعادل از طریق تأمین پویا و مهاجرت ماشین مجازی (VM یا مهمان) در میان گره‌های فیزیکی استفاده می‌کند. ماشین‌های مجازی در اینترنت به انواع بسیاری از فعل و انفعالات که در معرض تکنولوژی مجازی‌سازی هستند می‌تواند کمک کند در حالی که فیلتر اطمینان درجه بالاتری از امنیت است. به‌طور خاص، مجازی‌سازی می‌تواند به عنوان یک جزء امنیتی استفاده شود. به عنوان مثال، ارائه‌ی نظارت بر ماشینهای مجازی، مدیریت آسان بر امنیت خوشه‌های پیچیده، مزارع سرور و زیرساخت محاسبات ابری. با این‌حال، فن‌آوری‌های مجازی همچنین نگرانی‌های بالقوه‌ی جدیدی را با توجه به امنیت ارائه می‌کنند، همانطور که در بخش 4 خواهیم دید.

مشارکتها: هدف از این مقاله،: (الف) بررسی مسائل مربوط به امنیت محاسبات ابری. (ب) ارائه‌ی یک راه‌حل برای مسائل فوق.

در این مقاله مسائل مربوط به امنیت ابر و مدل، تهدیدها و شناسایی الزامات اصلی یک سیستم حفاظتی را تجزیه و تحلیل می‌کنیم. به‌طور خاص، یک چارچوب معماری، سیستم پیشرفته‌ی حفاظت ابر (ACPS)، برای افزایش امنیت گره ابر ارائه می‌کنیم. ACPS براساس نتایج حاصل از KvmSec (لومباردی و دی پیترو، 2009) است و

KvmSma (لومباردی و دی پیترو، 2010) پسوند امنیتی نمونه‌ی اولیه از ماشین مجازی لینوکس است (KVM کومرانت، سال)، این نیز از معماری TCPS الهام گرفته است (لومباردی و دی پیترو، 2010). ACPS یک سیستم محافظت کامل برای ابرها است که به‌صورت شفاف بر قطعات ابر و تعامل با منابع محلی و راه دور برای محافظت و بهبود حملات، نظارت می‌کند.

در زیر ما نشان دادیم که چگونه ACPS می‌تواند به‌طور کامل مجازی‌سازی را به ارائه حفاظت در سیستم‌های ابر مانند Eucalyptus به‌کار ببرد (نورمی و همکاران، 2009) و (Openecp، 2010) و (انومالی، 2009). در واقع، OpenECP یک کد منبع به طور کامل باز است؛ که معماری و کد یکسانی را به اشتراک می‌گذارد. پیاده‌سازی نمونه‌ی اولیه ارایه شده است. اثربخشی و عملکرد آن تست شده است. نتایج نشان می‌دهد که پیشنهاد ما در برابر حملات انعطاف پذیر است و سربار آن در مقایسه با ویژگی‌های ارائه شده بسیار کوچک است.

یکی از نتایج اصلی تحقیق ما یک چارچوب است که یک سیستم حفاظت ابری برای پشتیبانی مجازی‌سازی در سراسر میزبان‌های فیزیکی از طریق اینترنت را فراهم می‌کند.

نقشه راه. باقی‌مانده‌ی این مقاله به شرح زیر است: بخش بعدی مربوط به بررسی کارهای مرتبط است. بخش 3 اطلاعات پس زمینه را فراهم می‌کند، در حالی که بخش 4 مسائل مربوط به امنیت ابر را طبقه‌بندی می‌کند. بخش 5 موارد مورد نیاز برای ACPS و معماری را توضیح می‌دهد. در بخش 6 جزئیات اجرا ارائه می‌شود، در حالی که اثر و عملکرد آن در بخش7 بحث شده است. در نهایت، بخش 8 برخی از نتیجه‌گیری‌ها را مورد بحث قرار می‌دهد.

2. کارهای مرتبط

اگر چه مسائل خصوصی در رایاتش ابری توسط پیرسون شرح داده شده است (2009)، اما امنیت محیط رایانش ابری در گذشته کمتر مورد بحث قرار گرفته است. برخی از مسائل جالب امنیت در سال 2009 توسط Siebenlist بحث شده است. یک مقاله جامع در قالب بررسی جامع راه‌کارهای امنیت به وسیله Cachin و همکارانش ارائه شده است. یک ابر امن برای ارزیابی ریسک به تازگی توسط ENISA در سال 2009 ارائه شده است. همچنین پیشنهادی توسط Armbrust و همکارانش ارائه شده است که ارزش خواندن را دارا می‌باشد. این مقالات نقطه

شروع کار ما شده است و ما برای تعریف مسئله و بیان مشکل و همچنین بهبود مشکلات این راه‌کارها به آن‌ها رجوع کرده‌ایم.

یک مرجع اساسی برای کار ما استفاده از کار Ristenpart در سال 2009 است. این کار نشان می‌دهد که برای نمونه‌های افزایشی ماشین‌های مجازی، می‌توان یک co-resident بر روی ماشین مجازی هدف یافت. پس از آنکه co-resident با موفقیت به دست آمد. حملات می‌توانند به صورت تئوری اطلاعات هدف را استخراج کنند. هر مخربی ممکن است به طور متناوب نمونه‌های فعالی را ایجاد نماید. Ristenpart نشان می‌دهد که امکان اجاره‌ی ماشین‌های مجازی بیشتر برای داشتن co-resident وجود دارد و همچنین نشان می‌دهد که طراحی هر co-resident بسیار ساده است.

اغلب سیستم‌های یکپارچه و تشخیص نفوذ فعلی می‌توانند با موفقیت در محیط رایاتش ابری اجرا شوند. ابزارهای یکپارچه فایل سیستم و تشخیص نفوذ مثل Tripwire (Kim and Spafford در سال 1994) و AIDE (AIDEteam) در سال 2005) می‌توانند بر روی ماشین‌های مجازی توسعه یابند. اما در معرض حملات احتمالی کاربران مخرب هستند. علاوه بر آن، هنگامی که یک مخرب می‌فهمد که ماشین هدف بر روی یک ماشین مجازی قرار دارد ممکن است برای فرار از محیط مجازی، بر روی مولفه مونیتورینگ ماشین مجازی (VMM) آسیبی را ایجاد کند. اغلب راه‌کارهای جاری از ویژگی‌های VMM ها برای افزایش امنیت استفاده می‌کنند. درون گرایی مجازی یا همان Virtual introspection (Jiang et al در سال 2007) فرآیندب است که امکان مشاده وضعیت ماشین مجازی در از طریق VMM به وجود می‌آورد. SecVisor (Seshadri در سال 2008) و Lares (Payne et al.) در سال 2008) و KVM-L14 (Peter et al در سال 2009) نام چند روش برای مشاده یکپارچگی kernel است. Nickle (Riley et al.) در سال 2008) با مونیتورینگ kernel code توانستند kernel rootkits یا روتکیت های هسته را کشف کنند. به هرحال، Nickle نتوانست در برابر حملات data kernel مقاومت کند (Rhee et al) در سال 2009). اما راه‌کار ما می‌تواند.

جدول ۱: مقایسه ویژگی‌های ارائه شده توسط ACPS، TCPS، KvmSma (KSma) و KvmSec(KSec)

Feature	KSec	KSma	TCPS	ACPS
Semantic View	N	Y	Y	Y
Guest Component	Y	N	N	N
Transparency	N	Y	Part.	Full
Non-Blocking	Y	Y	Y	Y
SWADR	N	N	N	Y
Hot Recovery (by Replacement)	N	N	N	Y
Accountability	N	N	N	Y

بیشتر طرح های ارائه شده محدودیت‌هایی دارند که آن‌ها را از مورد استفاده بودن در سناریوهای توزیع شده منع می‌کند (مثل Secvisor که تنها از یک VM به ازای هر هاست حمایت می‌کند). چنین سیستم‌هایی غالبا، نیازمندی‌های خاص یاسیستم‌های توزیع شده را بررسی نمی‌کند. برای مثال، KVM-L4 تکنولوژی هایی را شبیه به کار (Lombardi و Di در سال 2009) استفاده می‌کند اما موارد اضافی در مورد محیط‌های پیچیده را مورد بررسی قرار نمی‌دهد. همچنین IBMon (Ranadive در سال 2009) از یک ابزار مانیتورینگ برای مانیتورکردن ناهمزمان یا مجازی سازی تجهیزات شبکه استفاده می‌کند. LoGrid(Salza در سال 2006) یک نمونه از سیستم تعاملی اتوماتیک است.

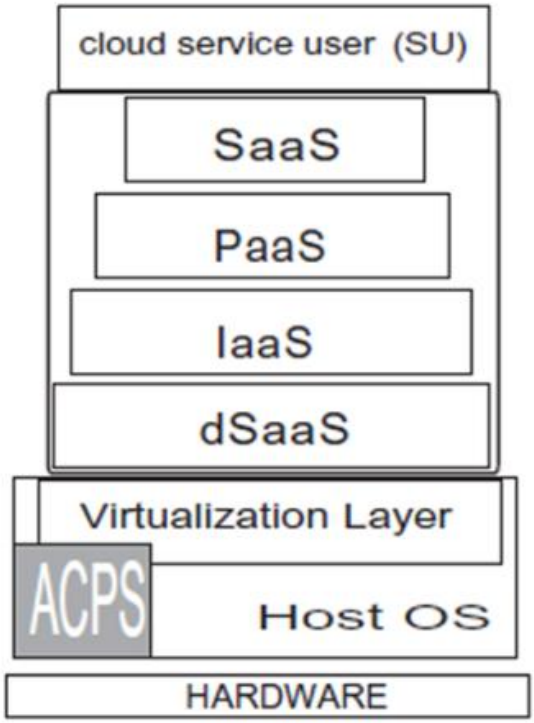
در تلاشی برای ایجاد انعطاف پذیری طولانی مدت در حملات گره‌ها، تحمل نفوذ Self-Claening (SCIT) (Huangدر سال 2006) برای تمامی سرورها به طور بالقوه لحاظ شد (وقتی حملات کشف نشده خطرناک هستند). SCIT سرورها را به صورت منظم بازیابی می‌کند. عیب چنین سیستمی آن است که از اکثر برنامه‌ها به صورت طولانی مدت پشتیبانی نمی‌کند. مشابه با آن VM-FIT (Distler در سال 2008) سرورهای کپی شده از هم را ایجاد می‌کند که می‌تواند به صورت دوره‌ای انعطاف‌پذیری سرورها را افزایش دهد. در نهایت روش (Sousa در سال 2007) بازیابی proactive را با سرویس‌هایی که اجازه‌ی تکثیر صحیح از روی سرورها را می‌دهد استفاده می‌کند. همراه با مزایای بسیار زیادی که برای مجازی سازی آورده شده است. چالش‌های دیگری هم برای مجازی سازی ارائه شده است.

در نهایت نویسندگان این مقاله در کاری دیگر سیستم حفاظت ابر شفاف (TCPS) را ارائه دادند که به عنوان یک پوستر در SAC 10 نشان داده شده است. این پوستر برخی از سناریوها و نیازمندی‌های رایج در ACPS را معرفی می‌کند. ACPS و TCPS شرایط مانیتورینگ سیستم را share می‌کنند. ACPS معماری کامل شده‌ی

TCPS است. برای مثال ACPS ویژگی‌های منحصربه فرد جالبی دارد مثل روش SWADR. تمام این ویژگی‌های جدید بر روی هر دو امنیت و کارایی تاثیر دارند و می‌توانند برای ساختن ایده‌های جدید مورد استفاده قرار گیرند (جدول 1 را ببینید).

3. پیش زمینه

ابر (Vaquero و همکاران، 2009) یک استخر از منابع مجازی در سراسر اینترنت است که به دنبال یک مدل پرداخت به ازای هر استفاده است و می تواند به صورت پویا پیکربندی مجدد گردد تا بتواند درخواست کاربر را برآورده سازد. محاسبات ابری یک مدل سرویس برای تامین نیازمندی‌های IT است که غالبا بر اساس تکنولوژی مجازی سازی و محاسبات توزیع شده و محاسبات بهره‌وری است (Lenk و همکارانش در سال 2009). روش‌های محاسبات ابری برای محاسبات توزیع شده استفاده از ایده محاسبات توری و محاسبات توزیع شده است. تفاوت اهداف این دو تکنولوژی در روش پیاده‌سازی و تمرکز آن‌ها است. از یک جهت با توجه به تکنولوژی گرید، کاربران محاسبات ابری کنترل کمی روی مکان داده و محاسبات دارند. به عبارت دیگر هزینه مدیریت رایانش ابری کمتر و مدیریت کمتر دست و پاگیر است. در ادامه به مولفه‌های زیرساخت ابری به عنوان middleware می‌پردازیم.



شکل ۱: لایه‌های ابر و سیستم‌های پیشرفته‌ی حفاظت از ابر

سرویس‌های ابری در لایه‌های متفاوتی ارائه می‌شوند (* به عنوان یک سرویس در شکل 1 نشان داده شده است) در حالت SaaS، ذخیره سازی داده به عنوان سرویس، ظرفیت ذخیره‌سازی پایه را بر روی شبکه ارائه می‌دهد. در حالت IaaS ، زیرساخت به عنوان سرویس ارائه شده و زیرساخت در قالب یک سخت‌افزار مجازی بدون هیچ نرم‌افزاری ارائه می‌شود. PaaS ، یک سرور مجازی یا سیستم عامل و کاربرد مجازی را در قالب سرویس به کاربر ارائه می‌دهد. SaaS، دستیابی به نرم‌افزار بر روی اینترنت را به عنوان یک سرویس امکان‌پذیر می‌سازد.

در این کار، تلاش‌ها بر روی لایه محاسباتی تمرکز دارند (مثل IaaS). بنابراین ما می‌توانیم چاچوب امنی را بر روی سرویس‌ها ارائه دهیم. (حتی اگر رابط‌های برنامه کاربردی باز و شناخته شده باشند.) این دلیلی است که چرا ما مدل اکالیپتوس و openECP را را برای پیاده‌سازی انتخاب کرده‌ایم. در ادامه ما بر روی مسئله امنیت تمرکز خواهیم کرد.

4. مسائل امنیتی ابر

یکی از مسائل کلیدی محاسبات ابری (نگاه کنید به شکل 1) از دست دادن کنترل است. به عنوان یک مثال اول، کاربر خدمات (SU) نمی‌داند که اطلاعات آن در ابر دقیقا کجا ذخیره و پردازش می‌شود. محاسبات و داده‌های تلفن همراه است و می‌تواند به سیستم‌هایی مهاجرت کند که SU قادر به کنترل مستقیم آن نیست. بر روی اینترنت، عبور از مرزهای بین‌المللی برای داده‌ها رایگان است و این می‌تواند باعث افزایش تهدیدات امنیتی شود. مثال دوم از دست دادن کنترل این است که ارائه دهنده‌ی ابر (CP) می شود برای اجرای یک سرویسی که جزئیاتش را نمی‌داند هزینه پرداخت می‌کند. این، یک قسمت تاریک از مدل "زیرساخت به عنوان سرویس" از دیگر رویکردهای یک سرویس است. تا به امروز، مشکلات تمایل دارند که با یک قرارداد خدمات کار کنند، که در آن توافق باید توسط ابزار کنترل اجرا و نظارت شود (هابرلین، 2009). برخی از مسائل امنیتی ابر در زیر آمده است (فاستر و همکاران، 2009):

SEI 1 دسترسی امتیاز کاربر: دسترسی به برون‌سپاری حساس داده‌ها به یک زیر مجموعه از کاربران ممتاز محدود شود(برای کاهش خطر سوء استفاده از نقش‌های با امتیاز بالا).

2SEI تفکیک داده: یک نمونه از داده‌های مشتری به‌طور کامل از داده‌های مشتری دیگر تفکیک شود.

SEI 3 حفظ حریم خصوصی: قرار گرفتن در معرض اطلاعات حساس ذخیره شده در سیستم عامل دلالت بر مسئولیت قانونی و از دست دادن شهرت دارد؛

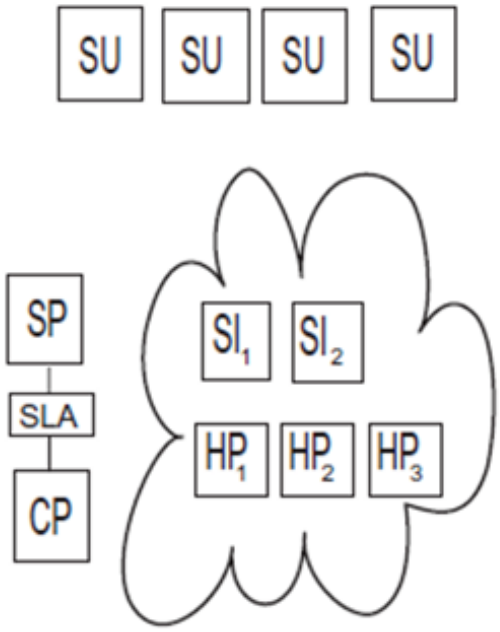
SEI 4بهره‌برداری باگ: مهاجم می‌تواند از یک باگ نرم افزار برای سرقت اطلاعات با ارزش بهره‌برداری کند و یا به حملات بیشتر به منابع اجازه دهد.

SEI 5 بازیابی: ارائه‌دهنده‌ی ابر یک تکرار کارآمد و مکانیزم بازیابی برای بازگرداندن خدمات ارائه می‌دهد.

6SEI پاسخگویی: حتی اگر خدمات ابر برای ردیابی جهت مقاصد پاسخگویی دشوار باشد، در برخی موارد یک نرم‌افزار الزامی است.

با توجه به نکته‌ی دوم، افزایش پاسخگویی می‌تواند باعث افزایش امنیت و کاهش ریسک برای کاربر سرویس و ارائه‌دهنده‌ی سرویس باشد. یک مبادله بین حریم خصوصی و پاسخگویی وجود دارد، زیرا رکوردی از اقدامات تولید می‌شود که می‌تواند توسط یک شخص ثالث زمانی که چیزی را اشتباه می‌کند مورد بررسی قرار گیرد. این بررسی‌ها ممکن است قطعات معیوب و یا منابع داخلی ابر را با جزئیات پیکربندی نشان دهد. به این ترتیب، یک مشتری ابر ممکن است قادر به یادگیری اطلاعات در مورد ساختار داخلی ابر باشد که می‌تواند برای انجام یک حمله مورد استفاده قرار گیرد. یک راه‌حل ممکن می‌تواند استفاده از ابهام و تکنیک‌های حفظ حریم خصوصی باشد تا VM اطلاعاتی محدود به ابر نشان دهد (بتنکورت و همکاران، 2009). به‌هرحال، تکنولوژی فعلی نمی‌تواند از دسترسی حافظه‌ی مهمان VMM جلوگیری کند. این برگ محرمانگی را با توجه به ارائه‌دهنده‌ی خدمات باز می‌کند (با توجه به یک مهاجم اگر او پلتفرم میزبانی را به خطر انداخته باشد).

4.1. مدل امنیت ابر



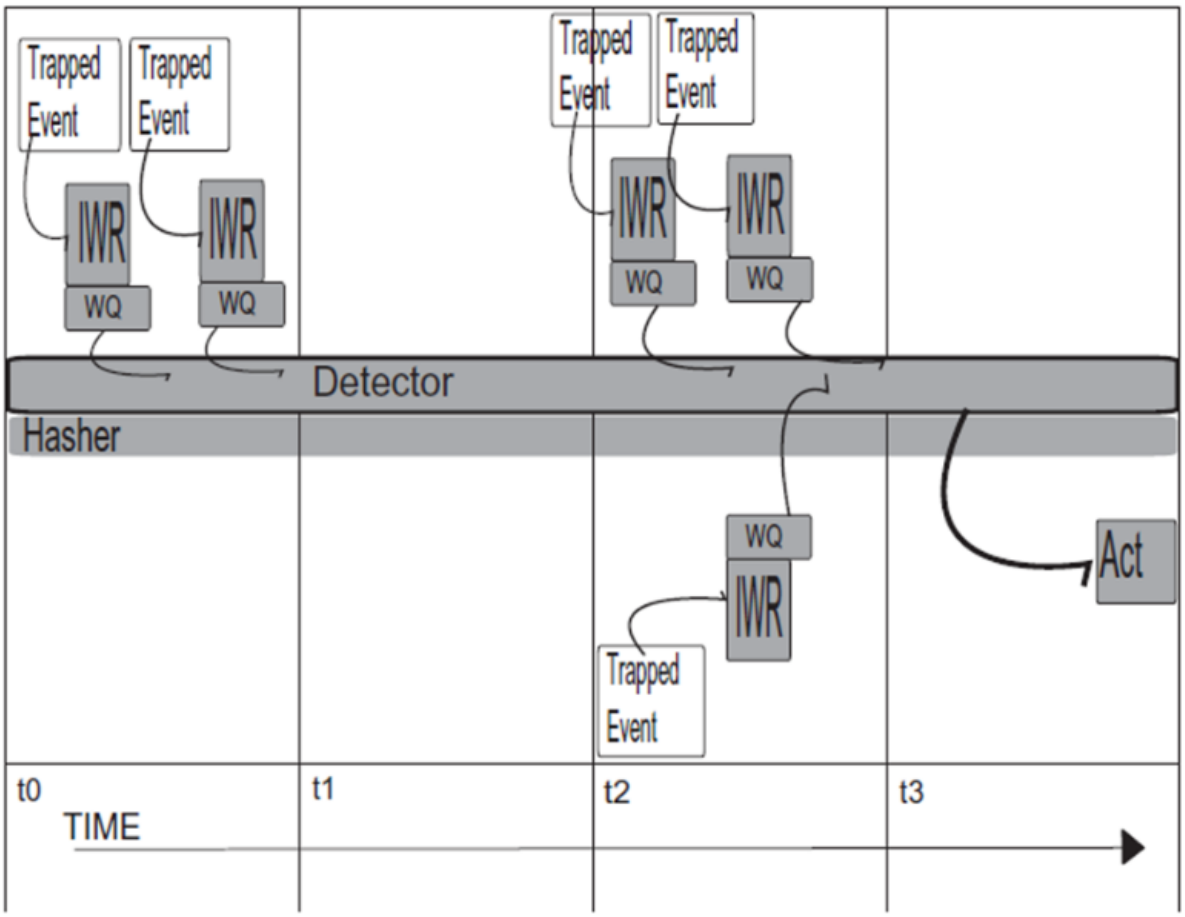
شکل ۲: . اجزای مدل سرویس ابر

شکل2 که سناریوی ارائه شده در این مقاله را نشان می‌دهد. ارائه‌دهنده‌ی خدمات

(SP) یک یا چند نمونه از خدمات (SI) را در ابر اجرا می‌کند، که می‌تواند توسط یک گروه از کاربران نهایی (SU) دیده شود. برای این منظور، SP منابعی از ارائه‌دهنده‌ی ابر (CS) را کرایه می‌کند. این است که SU و SP هیچ‌گونه کنترل فیزیکی بر روی

ماشین ابر، که وضعیتش قابل مشاهده نیست، ندارند. SU و CP به یک سطح سرویس وارد میشوند که توضیح می‌دهد چگونه ابر سرویس SI را اجرا خواهد کرد.

شکل ۳: نظارت گردش کار SWADR: رهگیر و ضبط هشدار(IWR)، هشدار استخر (WP)، محرک (قانون) فعل و انفعالات در طول زمان.



حملات ممکن در برابر سیستم‌های ابر می‌توانند به‌صورت زیر طبقه‌بندی شوند (اسمیت و همکاران، 2006):

1CAT حملات منابع برابر CP ها؛

2CAT حملات منابع در برابر SP ها؛

3CAT حملات داده‌ها در برابر CP ها؛

4CAT حملات داده‌ها در برابر SP ها؛

5CAT حملات داده‌ها در برابر SU ها.

حملات منابع (2-CAT1CAT) از نظر سوء استفاده از منابع، مانند سرقت منابع مجازی برای سوار شدن یک botnet در مقیاس بزرگ حمله است. حملات داده‌ها (3CAT، 4CAT) سرقت یا تغییر خدمات و یا گره پیکربندی داده‌ها است (که می‌تواند بعدا برای انجام یک حمله مورد استفاده قرار گیرد). حملات داده‌ها در برابر کاربران سرویس (5CAT) می‌تواند به نشت اطلاعات حساس منجر شود. کلاس‌های حمله‌ی 1CAT و 3CAT شامل یک

حمله به اجزای زیرساخت ابر هستند. فن‌آوری‌های مجازی‌سازی تحت زیرساخت‌های محاسبات ابری می‌توانند امنیت را به چالش بکشند (سکونیا، 2009). علاوه بر این، محاسبات ابری به طور بالقوه به برخی از حملات که هنوز مشخص نشده است اجازه می‌دهد. بعدا خواهیم دید که چگونه معاملات ACPS با چنین تهدیدهایی روبرو می‌شود.

5. سیستم پیشرفته‌ی محافظت از ابر

سیستم پیشرفته‌ی حفاظت از ابر پیشنهاد شده (ACPS) به طور فعال به حفاظت از یکپارچگی مهمان VM ها می‌پردازد و از میان محاسبات توزیع شده، به میزبان اجازه‌ی نظارت ماشین‌های مجازی مهمان و اجزای زیرساخت‌هارا می‌دهد. پیشنهاد ما روش KvmSec را (لومباردی و دی‌پیترو، 2009) برای محافظت از اجزای نظارت در برابر مزاحمان و حملات مانند کرم‌ها و ویروس‌ها گسترش داده است.

ACPS صرفا میزبان معماری سمت کاربر جهت اعمال نفوذ مجازی است (هی و نانسی، 2008). این اجازه می‌دهد به: استقرار هر وسیله‌ی مجازی مهمان " همانگونه که هست"؛ به اجرا درآوردن برخی از پاسخگویی‌ها در فعالیت مهمان بدون توجه به مهاجم واقع در مهمان. این ویژگی دوم به عنوان سیستم حفاظت سخت برای شناسایی ارائه شده است. در ادامه، مدل تهدید ACPS و الزامات مورد نیاز برای توصیف مختلف سیستم عامل محاسبات توزیع‌شده شرح داده شده است. پس از آن جزئیات اجرا همانند بررسی عملکرد اثر ACPS و پیاده‌سازی و طراحی و معماری محاسبات ابری بیان می‌شود.

5.1. مدل تهدید

در مدل ارائه شده، می‌توانیم بر تمامیت گروه‌ها تکیه کنیم، زیرا فرض بر این است که میزبان بخشی از محاسبات پایه‌ی مورد اعتماد (TCB) است (هومود و همکاران، 2004). زمانی که تصویر VM توسط یک نهاد مورد اعتماد ارائه می‌شود، صداقت مهمان در زمان راه‌اندازی با فرض در معرض تهدید قرار گرفتن VM مستقر شده، است. در واقع، مهمان می‌تواند هدف حملاتی از نوع سایبری و نفوذ به شبکه مانند ویروس‌ها، تزریق کد، و سرریز بافر شود. در مورد تصویر ارائه شده‌ی مهمان توسط کاربر، اعتماد VM را نمی‌توان تضمین کرد و اقدامات مهمان باید برای

ردیابی مخرب احتمالی فعالیت‌ها بررسی شود. در مدل ارائه شده، مهاجمان می‌تواند کاربران ابر (SP) و یا برنامه‌های کاربردی کاربران ابر (SU) باشند، در حالی که قربانیان می‌تواند ارائه‌دهندگان خدمات در حال اجرا در ابر (2CAT، 4CAT)، زیرساخت‌های خود ابر (1CAT، 3CAT) و یا کاربران دیگر (5CAT) باشد. تهدید سنتی زمانی است که یک مهاجم سعی در اجرای بهره‌برداری از راه دور از آسیب‌پذیری های نرم‌افزار در سیستم مهمان (2CAT) دارد. برخی از حملات با استفاده از خدمات ممکن ابر ساخته شده است (2-CAT1CAT)، زیرا یک بدافزار از نظر قانونی می‌تواند موارد دیگر در درون ابر را همانطور که قبلا مشخص شده است به‌کار ببرد، همچنین می‌تواند یادگیری اطلاعات محرمانه را مدیریت کند (5CAT) (مراجعه کنید به 2009).

5.2. نیازمندی‌ها

مجموعه‌ای از الزامات را که توسط یک سیستم مانیتورینگ امنیتی برای ابرها مورد نیاز است به‌صورت زیر شرح می‌دهیم (لومباردی و دی‌پیترو، 2009؛ لومباردی و دی‌پیترو، 2010):

1REQاثربخشی: سیستم باید قادر به تشخیص انواع حملات و نقض تمامیت باشد.

2REQ دقت: سیستم باید قادر به (ایده‌آل) اجتناب از مثبت_کاذب باشد. که در آن تشخیص نرم‌افزارهای مخرب حملات که در آن فعالیت‌های مجاز انجام می‌شود، اشتباه است.

3REQ شفافیت: سیستم باید دید VM ها را به حداقل برساند: SP، SU، و مزاحمان بالقوه باید قادر به تشخیص وجود نظارت بر سیستم.

subvertability 4REQ: سیستم میزبان، زیرساخت‌های ابر و اجداد مجازی باید از حملاتی که توسط یک مهمان به خطر بیافتد محافظت کنند و آن نباید امکان غیرفعال کردن و یا تغییر نظارت بر سیستم را داشته باشد.

Deployability 5REQ: سیستم باید در اکثریت قریب به اتفاق در میان‌افزار ابر و تنظیمات HW/SW گسترش یابد.

6REQ واکنش پویا: سیستم باید یک تلاش برای نفوذ در ابر را تشخیص دهد و اگر سیاست امنیتی لازم بود، باید اقدامات مناسب را در برابر تلاش و در برابر مهمان به خطر افتاده و/یا کنترل از راه دور میان‌افزار مدیریت امنیت اجزاء، انجام دهد.

7REQپاسخگویی: سیستم نباید با ابر و اقدامات نرم افزار ابر مقابله کند، اما داده‌ها و عکس‌های فوری برای اجرای سیاست‌های پاسخگویی باید جمع‌آوری شود.

یک تجارت بین شفافیت و عکس‌العمل پویا وجود دارد؛ ما این مشکل با مجموعه‌ای از راه‌حل‌های واکنشی ممکن ACPS با زیرمجموعه‌ی قابلیت تعمیر و نگهداری به‌طور منظم مهمان حل می‌شود، به عنوان مثال، توقف مهمان و راه‌اندازی مجدد آن از یک تصویر جدید و مهاجرت نمونه‌یVM. اقدامات فوق، از نقطه نظر SU یا SP، تقریبا از عملیات تعادل بار به‌طور منظم بر اساس VM غیر قابل تشخیص است.

5.3. روش ارائه شده

ما بر اجزای کلیدی نظارت می‌کنیم که می‌توان توسط حملات به منظور حفاظت از ماشین‌های مجازی و ابر تحت تاثیر زیرساخت، مورد هدف قرار داده شود. توسط دو کلید فعالانه یا منفعلانه نظارت و اجزای میان‌افزار، قادر به تشخیص هر گونه اصلاح ممکن در اطلاعات هسته و کد هستیم، در نتیجه تضمین آن هسته و یکپارچگی میان‌افزار به خطر نافتاده است. علاوه‌براین، به منظور نظارت بر نقاط ورود ابر، رفتار و یکپارچگی اجزای ابر از طریق ورود به سیستم و تأیید کنترل‌های دوره‌ای از فایل‌های اجرایی و کتابخانه‌ها بررسی می‌شود. بیشترین هدفی که ما در پی رسیدن به آن هستیم، به‌ویژه هنگامی‌که تصویر مهمان توسط ارائه‌دهنده‌ی ابر قابل اعتماد نیست، اطمینان از این است که نرم‌افزار مهاجم نمی‌تواند یک نفوذ خارجی در محل سیستم را تشخیص دهد. توجه داشته باشید که، به عنوان درون‌نگری تکنیک، هنوز مشخص نیست که تا چه حد می‌توان آنها را توسط ماشین مجازی هدف شناسایی کرد. در واقع، حضور یک نظارت بالقوه بر سیستم می‌تواند از طریق اندازه‌گیری زمان اجرای آن برای فراخوانی تابع خاصی تشخیص داده شود. با مشاهده‌ی این اعمال نفوذ، سیستم مانیتورینگ ما به‌طوری است که SWADR-خطر همزمانی- تشخیص ناهمزمانی و پاسخگویی را می‌توانیم تعریف کنیم. به‌طورخاص، ACPS می تواند حفاظت را ارائه کند:

1PRT از حملاتی که از خارج ابر می‌آیند.

2PRT از حملاتی که از هم‌نیاهای مجازی می‌آید؛

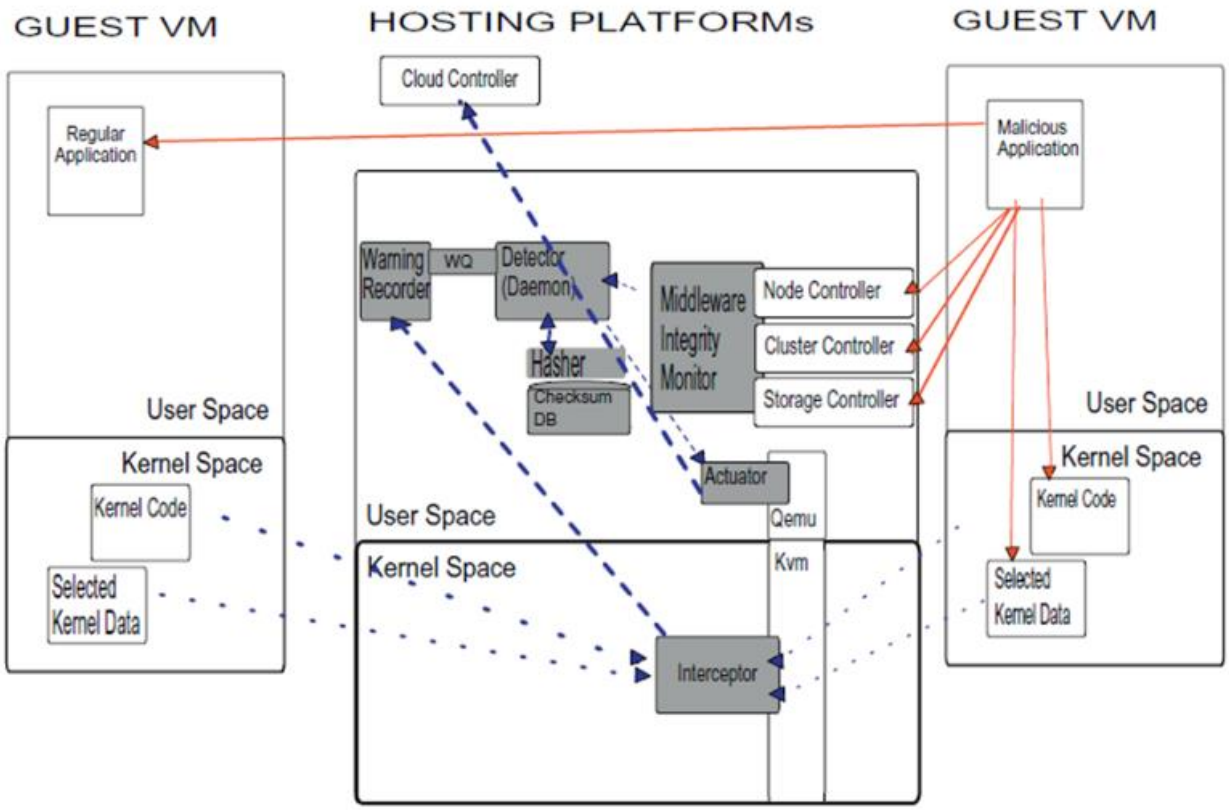
3PRT از حملاتی که از ماشین‌های مجازی می‌آید.

توضیحات سطح بالایی از ACPS همراه با اکالیپتوس و ACPS ترکیب شده با معماری OpenECP، در شکل 4 و 5 نشان داده شده است، به‌ترتیب، جریان داده‌ی خطرناکی که به‌طور بالقوه در خطوط پیوسته به تصویر کشیده شده و نظارت بر جریان داده‌ها با خط‌چین نشان داده شده است. تمام ماژول‌های ACPS در میزبان واقع شده است. ACPS از QEMU (بلارد، 2005) برای دسترسی به مهمان استفاده می‌کند. فعالیت‌های مشکوک مهمان (به عنوان مثال فراخوانی‌های سیستم) می‌تواند توسط رهگیر متوجه شده و توسط ضبط هشدار به استخر اخطار ضبط شود، که در آن تهدید بالقوه توسط مولفه ارزیاب ارزیابی خواهد شد شد. رهگیر تصوری برای جلوگیری یا رد هر گونه تماس سیستم، به منظور جلوگیری از نظارت سیستم جهت شناسایی ندارد: در حالت SWADR، زمان حمله خنثی شده است. در واقع، اجزای ارزیابی (ارزیاب و Hasher) همیشه فعال -شکل 3-در حال اجرا و به‌طور مداوم درحال انجام چک‌های امنیتی هستند. در واقع، ارزیاب و Hasher حتی زمانی که استخر هشدار خالی است فعال و در حال اجرا هستند. در این مورد، هدف از استخر هشدار به طور عمده خفه کردن جزء ارزیابی نیست. استخر اخطار همچنین تعیین اولویت به منظور ارزیابی را اجازه می‌دهد. همچنین افزایش نامرئی را تضمین می‌کند ، حتی اگر تعداد زیادی از هشدارهای ممکن به طور بالقوه موجب تاخیر تصمیم و واکنش‌های محرک شوند. با توجه به چنین مسئله‌ای، افزایش نرخ هشدارهای دریافتی می‌تواند به عنوان یک تهدید امنیتی باشد. درست است که، در روش آسنکرونSWADR مهاجم برای انجام برخی از دستکاری‌ها در زمان محدود با سیستم هدف می‌تواند اجازه دهد. این هم درست است که به منظور انجام تغییرات به سیستم مهمان، نفوذگر باید درحال‌حاضر مشغول کنترل چنین سیستمی باشد. علاوه براین،undetectability نظارت بر سیستم اجازه می‌دهد تا رفتارهای نرم‌افزار مخرب در یک Honeypot مشاهده شود.

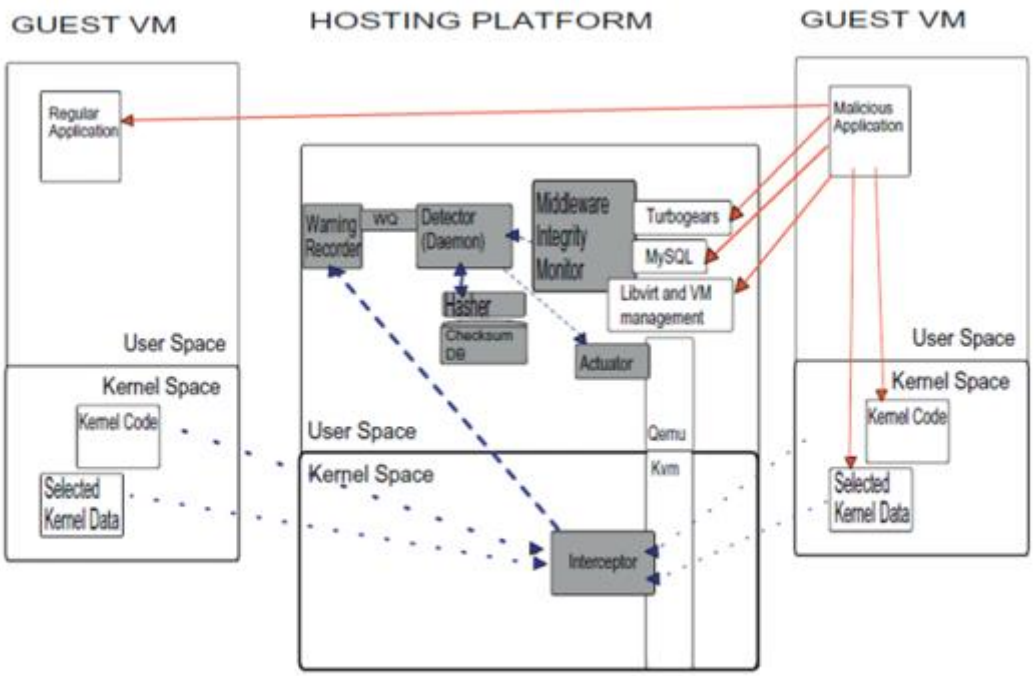
ACPS از ویژگی‌های زیر برخوردار است: برای ماشین مهمان شفاف است (حتی آنهایی که مخرب و یا غیرقابل اطمینان هستند). از مجازی‌سازی کامل پشتیبانی می‌کند (پرز و همکاران، 2008)، که سیستم در سمت مهمان کمتر قابل تشخیص باشد و می‌تواند در بسیاری از معماری‌های مبتنی بر 86x و X 86_64 در سیستم عامل‌های توزیع محاسبات ابری مستقر شود.

ACPS تفاوت قابل توجهی بر نظارت سیستم‌های امنیتی ارائه شده در KvmSec (لومباردی و دی پیترو، 2009) و KvmSma (لومباردی و دی پیترو، 2010) دارد؛ برخی از تفاوت‌های اصلی بین KvmSec، KvmSma، TCPS

و ACPS در جدول 1 نشان داده شده است. مهم‌ترین، ACPS به طور کامل به ماشین مهمان، ویژگی‌های حالتSWADR (3REQ را ببینید)، استخر هشدار و امکان بازیابی جایگزینی برای سرویس به خطر افتاده شفاف است. مصالحه حتی از یک (درحال‌حاضر) ماشین مجازی به خطر افتاده و غیر قابل اطمینان دشوار است(1CAT و 3CAT را ببینید)، در حالی‌که آن می‌تواند به‌صورت شفاف سمت مهمان را بازرسی و داده‌ها را تجزیه و تحلیل کند. ACPS از پاسخگویی همانطور که در این بخش بحث شد حمایت می‌کند (7REQ را ببینید) و اجازه‌ی ردیابی و ثبت وضعیت مهمان و داده‌های گرفته شده از طریق عکس‌های فوری را می‌دهد، بنابراین از تجزیه و تحلیل کامل در میان ابرهای یکپارچه‌ی موجود برخوردار است. ACPS، همانند TCPS، به‌طورکامل بر روی دستگاه میزبان (3REQ را ببینید) واقع شده است. در ACPS هر ماشین مجازی از منطقه‌ی حافظه خصوصی خود استفاده می‌کند، بنابراین کاملا از دیگر ماشین‌های مجازی (4REQ و 4CAT را ببینید) مستقل است.



شکل ۴: ACPS (قطعات به رنگ خاکستری) همراه با معماری اکالیپتوس



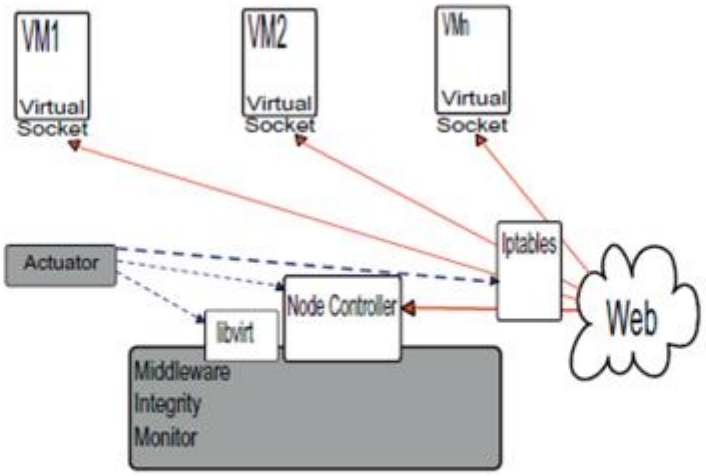
شکل 5: ACPS(قطعات به رنگ خاکستری) همراه با معماری OpenECP

در ACPS، پایگاه داده‌ی سمت میزبان DB شامل چک سام محاسبه شده برای زیرساخت‌های حیاتی انتخاب شده توسط میزبان و کد هسته مهمان، داده‌ها و فایل‌ها است. در زمان هشدار ضبط اجرا می‌تواند به‌صورت ناهمگام مقادیر هش را برای چنین اشیایی دوباره محاسبه کند و فایل هشدار می‌تواند تحت نظارت ارزیاب باشد. ارزیاب هشدارها و ارزیابی‌ها را مورد بررسی قرار می‌دهد (2-REQ1REQ را ببینید) که آیا امنیت سیستم به خطر افتاده است. در چنین حالتی محرک است با استناد به سیاست امنیتی مشخص شده (6REQ) عمل می‌کند. در نتیجه، ACPS به‌صورت محلی می‌تواند به نقض امنیت و یا اطلاع لایه مدیریت امنیت برای چنین اجزای که حادثه رخ داده است واکنش نشان دهد. ACPS همچنین می‌تواند جایگزین یک سرور به خطر افتاده برای بازگرداندن VM از یک تصویر تمیز پشتیبان شود (دیستلر و همکاران را ببینید.، 2008). برای جلوگیری از مثبت کاذب به اندازه ممکن (2REQ)، یک مدیر و یا مولفه‌ی کنترل ابر می‌تواند ACPS چک سام اجزای جدید را مطلع سازد. ACPS در نرم‌افزار مجازی‌سازی و اهرم یکپارچه پشتیبانی سخت‌افزار مجازی‌سازی برای نظارت بر یکپارچگی مهمان و انجام کنترلی میان قطعات از جمله اشیاء است. شایان ذکر است که هیچ فراخوانی سیستمی که تاکنون مسدود شده و یا با تاخیر به ACPS برای بررسی واکنش سیستم مانیتورینگ ارائه نشده است، (انجماد/توقف/راه اندازی مجدد مهمان VM) عملا از وظایف تعمیر و نگهداری سیستم غیر قابل تشخیص است. علاوه براین، رهگیر و هشدار

می‌تواند وقایع، اقدامات و ردیابی که آنها انجام می‌دهند را ضبط کند. این داده‌ها، همراه با داده‌های کنترلی DB، می‌تواند برای اهداف پاسخگویی (7REQ) استفاده شوند(هابرلین، 2009). معماری ارائه شده با پشتیبانی لازم برای ثبت وقایع خارجی امن و ابزار پاسخگویی در اجرای عملیات فراهم شده است.

6. اجرا

ما ACPS را بر روی اکالیپتوس و OpenECP (5REQ) اجرا کردیم. اجزای سیستم سطح بالای اکالیپتوس به عنوان وب سرویس اجرا می‌شوند. اکالیپتوس (نورمی و همکاران، 2009) تشکیل شده است از: یک کنترل گره (NC) که اجرا، بازرسی و VM را بر روی میزبانی که در آن اجرا می‌شود کنترل می‌کند. کنترل خوشه (CC) که اطلاعاتی در مورد VM جمع‌آوری می‌کند و اجرای VM را بر روی کنترل گره خاص برنامه‌ریزی می‌کند؛ همچنین شبکه را به عنوان مثال مجازی مدیریت می‌کند؛ کنترل ذخیره‌سازی (SC) Walrus یک سرویس ذخیره‌سازی است که یک مکانیزم برای ذخیره‌سازی و دسترسی به تصاویر VM و داده‌های کاربر ارائه می‌کند؛ یک کنترل‌کننده‌ی ابر (CLC)، وب سرویس‌هایی های برای ورود کاربران و مدیران ارائه می‌کند که باعث تصمیمات برنامه‌ریزی سطح بالا می‌شود.



شکل 6: ACPS(قطعات به رنگ خاکستری) همراه با جزئیات اکالیپتوس

شرح جزئیات بیشتری از چگونگی ادغام ACPS با اجزء اکالیپتوس در شکل 6 گزارش شده است. در اکالیپتوس، ACPS می‌توان با کنترل ابر ، کنترل خوشه مستقر و مهمتر از همه، کنترل گره ادغام شود. NC قابل اجرا بر روی

هر گره میزبانی VM است. ما به‌خصوص بر فعالیت‌های NC و صداقت آن نظارت می‌کنیم، زیرا این جزء کلیدی برای اجرای ابر است (رلرمیر و همکاران، 2009). در واقع، همان‌گونه که در شکل6 نشان داده شده است، هنگامی‌که یک حمله یا یک تغییر به طور بالقوه خطرناک شناسایی می‌شود، محرک می‌تواند پیکربندی NC ، Libvirt و iptable را به منظور جلوگیری از خسارت بیشتر تغییر دهد. واکنش‌های ممکن شامل مهاجرت مهمانان که هیچ هشداری را نسبت به میزبان دیگر (مهمان تمیز)، درحالی‌که غیرفعال کردن گره میزبان مشکوک است افزایش نمی‌دهد.

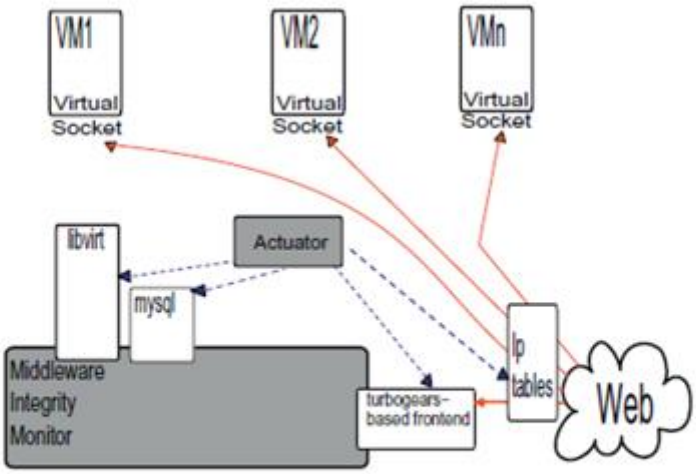
OpenECP ، مانند Enomalism اختصاصی خود (انومالی، 2009)، مقررات و مدیریت منابع با اعمال نفوذ، پایتون، و کتابخانه‌ی Libvirt (ردهت، 2007) است. اینها منابع زیرساخت‌های اضافی ما هستند که نیاز به نظارت برای یکپارچگی دارند. اجزای که باید بررسی شود پایتون، Libvirt و خروجی زیر فرآیندها، فایل‌های اجرایی و کتابخانه‌ها همانند فایل‌های پیکربندی هستند. اجزای جلویی Turbogears نیاز به نظارت ویژه دارند، زیرا به‌ویژه آنها در معرض شبکه هستند. چنین نظارتی یکپارچگی برای حفاظت از هر دو سیستم جلویی و پایانی را فراهم می‌کند (در برابرCAT 1). جزئیات ادغام Enomalism در شکل 7 نشان داده شده است. به‌طورخاص، در موردی که یک حمله یا یک تغییر به طور بالقوه خطرناک تشخیص داده شود ، محرک می‌تواند خروجی، Turbogears، Libvirt و پیکربندی iptables را به منظور جلوگیری از خسارت بیشتر تغییر دهد. واکنش ممکن شامل فیلتر کردن درخواست‌های وب انتخاب شده، مهاجرت مهمان تمیز به میزبان دیگر و ناتوانی گره میزبانی مشکوک است.

7. اثربخشی-ACPS مورد حمله

در این بخش نشان می‌دهیم که چگونه پیشنهاد ما از عهده‌ی حملات ابر در محیط‌های واقعی می‌تواند بر بیاید. به‌طورخاص، آزمایشهای عملی برای ارزیابی انجام انعطاف‌پذیری از معماری ارائه شده گزارش می‌کنیم و همچنین بحثی در مورد این‌که چگونه نیازهای کلیدی آورده شده در بخش‌های قبلی در پیشنهاد ما مشاهده شده است، ارائه می‌کنیم.

قابلیت‌های تشخیص (جدول3) سیستم ما ارزیابی در برابر تکنیک‌های حمله‌ی شناخته شده است (جدول2). اما، از آنجاکه کد منبع برای بسیاری از حملات در دسترس عموم نیست، تست ما با شبیه‌سازی مراحل حمله انجام می‌شود.

همانطور که قبلا هم نشان داده شده است، ما می‌توانیم حملات را به 5 دسته تقسیم کنیم، اعم از 1CAT تا 5CAT. ACPS برای شناسایی ثابت شده است ودر واکنش به حملات متعلق به دسته‌بندی‌ها که در بالا ذکر شد، در جدول 3 آمده است. به‌طورخاص خلاصه، ما از گذشته‌ها و برخی از حملات مرتبط که شبکه واقعی معماری می‌تواند موضوع را به خوبی نشان دهد استفاده کرده‌ایم (هوانگ و همکاران، 2007؛ ریستن پارت،2009؛ کوستا و همکاران، 2005) و ما درجه‌ی حفاظت اضافه شده توسط ACPS به مهمانان و VM زمانی که سیستم در معرض حملاتی قرار گرفته است نشان دادیم.



شکل 7: ACPS(قطعات به رنگ خاکستری) همراه با جزییات OpenECP

Category	Implemented Attack
CAT1	Apache vuln. (Eucalyptus)/ssh Python vuln. (OpenECP)
CAT2	Sebek rootkit
CAT3	network probing
CAT4	colocation, detection
CAT5	colocation, keystroke timing

جدول 2: نمونه حملات

به‌طورخاص، ما یک حمله از نوع 1CAT را توسط بهره‌برداری از آسیب‌پذیری‌های سرویس میزبان شبیه‌سازی کردیم (نگاه کنید به دبیان ،2008 و آسیب‌پذیری آپاچی CVE، 2008).

Attack Technique	Detection reason	Implemented Reaction
Apache/Python/ssh	process footprint	service migration/restart
Sebek	altered sys_call table	clean VM restart
Process Hiding	tasklist navigation	clean VM restart
colocation, network probing	Iptables monitoring	Silently filter/drop network packets

جدول 3: تشخیص قابلیت / واکنشACPS .

در این مورد ACPS بر رفتار پردازش Apache و حافظه نظارت می‌کند و متوجه استفاده غیر طبیعی از حافظه و تلاش برای اتصال است. پس از اینکه حمله را تشخیص داد، ACPS سرویس را از یک فایل اجرایی تایید شده مجدد راه‌اندازی می‌کند و پیکربندی فایل‌ها را دوباره ایجاد می‌کند.

ما حمله‌ی 2 CAT را با قرار دادن یک rootkit Sebek (پروژه هانینت. سبک، 2003) در یک پنجره VM انجام دادیم. Sebek یک ماژول کرنل است که حضور و رهگیری فایل سیستم و فعالیت‌های شبکه را پنهان می‌کند. این کار را با تغییر جدول فراخوانی سیستم به منظور تغییر جریان اجرا و برای اجرای کدهای مخرب انجام می‌دهد. در اینجا ACPS هر دو تغییر جدول فراخوانی سیستم و تغییر در کنترل فایل‌های هسته در ذخیره‌سازی مجازی را تشخیص می‌دهد.

ما همچنین یک حمله از نوع 4CAT را با استفاده از یک حمله‌ی هسته داده‌ها به صورت شرح داده شده در (ری و همکاران، 2009) اجرا کردیم. به‌طورخاص، با توجه به‌این‌که کارت شبکه توسط نرم‌افزار QEMU شبیه‌سازی شده است، مهمانان توسط ACPS از رویکرد حملات شبکه دروغ‌گو محافظت می‌شوند. در این زمینه، ما روش پنهان کردن فرآیند را اجرا کردیم که اجازه می‌دهد تا حملات را بدون نیاز به قرار گرفتن در لیستی از فرآیندها اجرا کند. این حمله با استفاده از یک حمله‌ی اعمال نفوذ داده‌های پویا برای دستکاری ساختار لیست کار انجام شده است. ACPS تغییر زمانی را با مرور لیست زمانبندی کار هسته تشخیص می‌دهد و درمی‌یابد که ساختار مخفی اضافی وجود دارد. به عنوان یک واکنش، ACPS مهمانی از یک دیسک VM را مجددا راه‌اندازی می‌کند.

در نهایت، ما حملاتی از نوع 3CAT و 5CAT را توسط استفاده از تکنیک‌های Ristenpart در(Ristenpart، 2009) اجرا می‌کنیم. اول از همه، هر دو شبکه‌ی خارجی (خارج از ابر) و داخلی (از هم‌نیای مجازی) از طریق اسکن پورت توسط قوانین iptables که یک هشدار را برای ضبط هشدار(WR) اجرا می‌کنند جلوگیری می‌شوند. در واقع، داشتن ACPS در حال اجرا در پردازنده‌ی تحت حمله باعث می‌شود زمان اندازه‌گیری برای مهاجم بسیار سخت‌تر شود.

7.1. آناتومی از حمله و واکنش

در زیر، جزئیات حمله‌ی نمونه که انجام دادیم توصیف شده است و واکنشی که ما از ACPS به‌دست آوردیم (میزبان و یکپارچگی سیستم‌های میهمان، فرض بر این است در زمان 0t اجرا شده است):

1. مهاجم (ATT) از آسیب‌پذیری SSH استفاده می‌کند و یک رمز عبور ضعیف برای دسترسی به حساب کاربری ایجاد می.
2. ACPS ، iptables را برای ضبط هشدار(WR)، تعداد و تلاش برای اتصال به SSH به‌کار می‌برد.
3. پس از آن ATT یک لینک حمله‌ی نمادین (جانستون، 2009) برای به‌دست آوردن دسترسی ریشه انجام می‌دهد.
4. ATT تکه کد فراخوانی بحرانی سیستم را درست می‌کند.
5. رهگیر ACPS متوجه عملیات بر روی شی هسته و فایل‌ها و اسناد است و در نتیجه چنین عملیات به طور بالقوه خطرناکی در WR رخ می‌دهد.
6. ارزیاب ACPS هشدارهای صادر شده توسط ضبط هشدار را بازخوانی می‌کند و یکپارچگی قطعات تحت تاثیر مقایسه‌ی چک‌سام را بررسی می‌کند.
7. هنگامی که یک هشدار برای مدیریت امنیتی از راه دور شناسایی شد؛ VM متوقف شده و دوباره آغاز می‌گردد(6REQ را ببینید).

ارزیاب ACPS می‌تواند به پیکربندی واکنش نشان دهند (به عنوان مثال راه‌اندازی یک راه‌اندازی مجدد سرویس) که شماره مورد نظر از حمله، جمع‌آوری شده است. افزایش امنیت ارائه شده توسط این رویکرد باید توسط افزایش

ممکن هر دو خدمات خرابی (DoS) و استفاده از منابع محاسبات که در موارد مثبت کاذب افزایش می‌یابد، متعادل شود.

7.2. کارآیی

در این بخش ما، نتایج حاصل از آزمایش با هدف ارزیابی عملکرد ACPS که در حال اجرا بر روی راه‌حل‌های ابر پیاده‌سازی می‌شوند، بیان می‌کنیم. پیکربندی سخت‌افزار/ نرم افزار برای این آزمون در جدول 4 به تصویب رسیده است. سیستم عامل‌های میهمان به تصویر کشیده86 با اعمال نفوذ 5.2Centos و یک CPU مجازی ویک گیگابایت رم است. مجازی‌سازی سخت‌افزار بر روی میزبان فعال شد. مهمان سیستم‌عامل‌های 32 بیتی را اجرا می‌کند درحالی‌که میزبان سیستم‌عامل 64 بیتی را اجرا می‌کند. دیسک مجازی مهمان از یک فایل تصویری ساخته شده بر روی میزبان استفاده می‌کند.

ما عملکرد راه‌حل‌مان را در سه مورد متفاوت مورد آزمایش قرار دادیم:

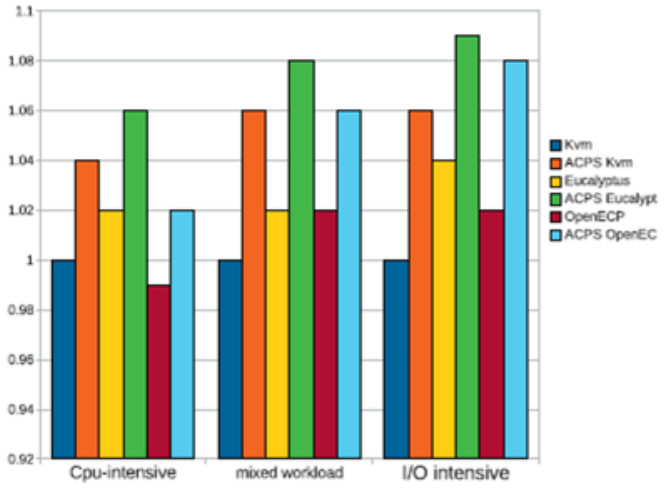
1. CPU فشرده؛
2. حجم کار مخلوط؛
3. I / O فشرده.

در جزئیات، ما اکالیپتوس و مهمان OpenECP را شرط کردیم و زمانی که آن مانند اکالیپتوس و OpenECP مهمان برای انجام سه نوع مختلف از عملیات طول می‌کشد اندازه‌گیری کردیم: کد کردن 3MP یک فایلwav ، هسته تلفیقی لینوکس 2.6.30 (حجم کار مخلوط) و dd از یک فایل بزرگ (1GB) برای پارتیشن‌بندی هارددیسک (I / O فشرده).

Feature	host A	host B
CPU Model	Athlon 64 4400+	Turion 64 RM-72
Cores	2	2
Ram	4096	4096
Host OS	Ubuntu 8.10 (O.ECP) Ubuntu 9.10 (Eucal.)	Ubuntu 8.10 (O.ECP) Ubuntu 9.10 (Eucal.)
Kernel	Linux 2.6.30	Linux 2.6.30
VMM	Kvm 88	Kvm 88

جدول 4: میزبانی محیط آزمون.

نتایج در شکل 8 گزارش شده است به‌طوری‌که میله‌ها نشان‌دهنده‌ی زمان اجرای نرمال همانند اجرای آزمون در ماشین KVM مهمان در میزبان یکسان است. ارزش‌ها بر روی پردازنده‌های تست شده به طور متوسط قرار دارند و نشان می‌دهد که سربار معرفی شده توسط ACPS بسیار کوچک است. یک کاهش عملکرد کوچک با توجه به یکپارچگی اضافی انجام شده ACPS در میان‌افزار ابر وجود دارد. تفاوت‌های بین نتایج Enomalism و اکالیپتوس را می‌توان با تفاوت در تعداد و پیچیدگی اجزای این دو توضیح داد. این معیار به ما برای تعیین کمیت واقعی سربار نرم‌افزار در دنیای واقعی معرفی شده توسط اجزای ناهمزمان نظارت کمک کرد. برای این منظور، میله‌ها باید دو به دو مقایسه شوند، نوار سمت چپ نشان دهنده‌ی کارآیی بدون ACPS است، درحالی‌که سمت راستی نشان‌دهنده‌ی کارآیی با ACPS فعال است. در واقع، تاثیر ACPS بر عملکرد فعلی راه‌حل‌های ابر کاملا محدود است، با توجه به اینکه حداکثر از دست دادن عملکرد زیر 6٪ است. به‌طورخاص، برای آزمون CPU فشرده می‌توان آن را کمتر از 3 درصد در نظر گرفت.



شکل 8: زمان اجرایACPS - برای بار اول

این نتایج تعجب‌آور نیست چرا که در SWADR، ارزیابی به عنوان یک فرآیند با اولویت پایین اجرا می‌شود و بیش از زمان گسترش می‌یابد، در نتیجه ترک منابع CPU در بیشتر قسمت‌ها رایگان است. نتایج پیچیده‌تر زمانی که به دنبال حجم کار مخلوط و نتایج فشرده‌ی I / O حجم کار هستیم، به‌دست می‌آید. این احتمالا با توجه به تعداد تعاملات ACPS فعال به علت فعالیت‌های سیستم فایل افزایش یافته است. بااین‌حال، تاثیر ACPS در عملکرد این نوع از حجم کار بیش از 6٪ نمی‌شود و به‌طور متوسط، نتایج ارائه شده بسیار جالب است.

ما پس از آن یک سری از آزمون‌ها را به منظور جمع‌آوری دقیق اندازه‌گیری عملکرد، با منافع خاص در زیر سیستم I / Oانجام دادیم. به‌طورخاص، آزمون‌ها منتخب زیر (اسمیت و همکاران، سال) اجرا شده است:

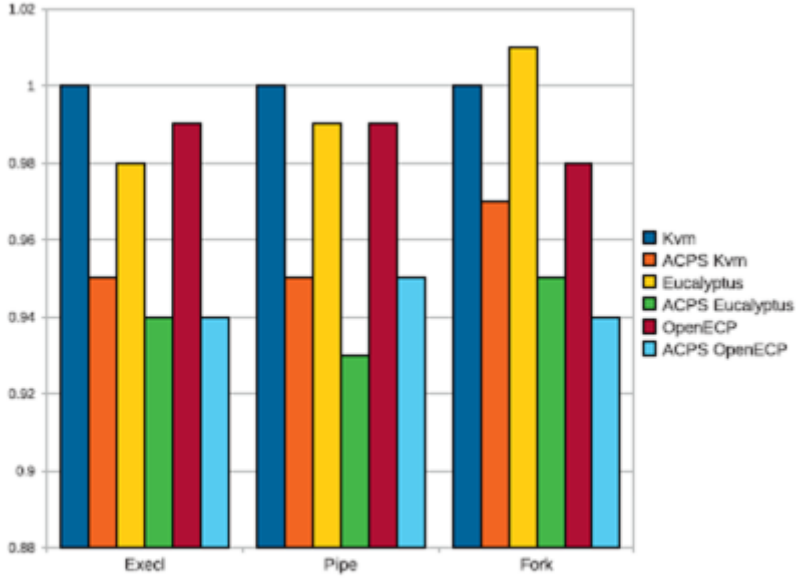
1. Execl : این تست تعداد فراخوانی‌های تابع execl () را که می‌تواند در یک ثانیه انجام شود اندازه‌گیری کرده است. Execl با هدف جایگزین کردن روند فعلی با یک روند جدید است. از این رو، این عملیات بر عملکرد حافظه‌ی I / O تاثیر می‌گذارد.

2. لوله: این تست تعداد لوله‌ها را می‌نویسد (512 بایت) یک فرایند موفقیت در یک ثانیه انجام می‌شود. و یک نشانه از سرعت روند در انجام I / O فعالیت‌ها است.

3. چنگال: این آزمون تعداد دفعات فراخوانی fork() در هر واحد از زمان را اندازه‌گیری می‌کند. این تست یک شاخص مهم از عملکرد کلی است.

نتایج در شکل 9 گزارش شده است. همانطور که در بالا، مقایسه دو به دو و با توجه به ستون انجام شده است. میله نشان‌دهنده‌ی تعداد اجرای عملیات است، از این رو یک نوار بالاتر به معنی کارایی بهتر است. این معیار برای تعیین کمیت سربار کارآیی عملیات خاصی با توجه به ACPS کمک می‌کند. خبر خوب این است که از دست دادن عملکرد با توجه به یکپارچگی اضافی قابل استفاده در هر آزمون ACPS کمتر از 6٪ است. در جزئیات، در این مورد عملکرد برای دو محیط محاسبات ابری بسیار مشابه است. دلیل این است که زیرساخت‌های محاسبات ابری به طور غیر مستقیم تحت تاثیر اجرای عملیات از جمله سطح پایین بستگی دارند، که خبر اجرایش به پیکربندی سیستم‌عامل و چک‌های مربوط به امنیت بستگی دارد (حتی اگر این آنهایی که نمی‌توان از حجم کار نرمال تشخیص داد). فعالیت ACPS عمدتا عملکرد I / O را تحت تاثیر قرار می‌دهد (مشاهده نتایج لوله‌های با عملکرد تا 6٪) در حالی

که از دست دادن عملکرد آزمایش fork کمتر از4٪. است چنین تفاوتی می‌تواند به دلیل تعامل با قطعات رهگیری ACPSباشد.



شکل 9: مقایسه عملکرد ACPS - برای بار دوم

نتایج نشان می‌دهد که یک حاشیه از بهبود برای نظارت عملیات I / O وجود دارد. این حاشیه‌ی بهبود را می‌توان توسط این واقعیت که اجرای I / O به طور کامل نظارت می‌شود توضیح داد. در واقع، نیاز به تعامل فوق العاده با زیر سیستم I / O است که می‌تواند در پیاده‌سازی‌های آینده‌ی چارچوب ACPS کاهش یابد. به‌طورکلی، این نتایج اولیه بنا به سربار معرفی شده کم است و ما را تشویق به تحقیق بیشتر به منظور اعمال نفوذ بهبود حاشیه که قبلا مشخص شده است می‌کند. در نهایت، اگر عملکرد کلی سیستم توسط نظارت خود تخریب شود، مانند مجازات عملکرد، نمی‌تواند توسط مهاجم از بار منظم پردازنده تشخیص داده شود، زیرا تفاوت بین تنظیمات زمان فراخوانی سیستم حفاظت شده و حفاظت نشده به طور مداوم در محدوده‌ی 3٪ -6٪ است، که تقریبا از دست دادن عملکرد با توجه به عملیات منظم غیر قابل تشخیص است.

8. نتیجه‌گیری

در این مقاله، ما کمک‌های زیادی برای فراهم کردن امنیت ابرها از طریق مجازی‌سازی انجام دادیم. ابتدا، یک معماری پیشرفته‌ی (ACPS) را برای محافظت از ابر پیشنهاد دادیم که می تواند بر هر دو مهمان و صداقت

میان‌افزار نظارت کند و از آنها در بسیاری از انواع حمله محافظت کند درحالی‌که هنوز به طور کامل برای کاربران خدمات و به ارائه‌دهنده‌ی خدمات شفاف نیست. ACPS بر روی پیاده‌سازی‌های مختلف ابر طراحی و مستقر شده است و قادر است به‌صورت محلی به نقض امنیت و آگاه ساختن لایه مدیریت امنیت هنگام وقوع واکنش نشان دهد. ثانیا، معماری ارائه شده به‌طور کامل در راه‌حل‌های متن‌باز اجرا شده است و هر دو اثر حفاظت و نتایج کارآیی، جمع‌آوری و تجزیه و تحلیل شده است. نتایج نشان می‌دهد که روش پیشنهادی موثر است و فقط یک پنالتی کوچک از کارآیی را معرفی می‌کند.

تشکر و قدردانی

ما می‌خواهم از داوران ناشناسی که با نظرات مفید خود در بهبود کیفیت مقاله به ما کمک کردند تشکر کنیم.

References

AIDEteam. Advanced intrusion detection environment. 〈http://sourceforge.net/projects/aide〉, November 2005.

Armbrust M, Fox A, Griffith R. Above the clouds: A Berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, February 2009.

Bellard F. Qemu, a fast and portable dynamic translator. In ATEC '05: Proceedings of the annual conference on USENIX annual technical conference, Berkeley, CA, USA, 2005. USENIX Association, p. 41.

Bethencourt J, Song D, Waters B. New techniques for private stream searching. ACM Trans. Inf. Syst. Secur. 2009;12(3):1–32.

Cachin C, Keidar I, Shraer A. Trusting the cloud. SIGACT News 2009;40(2):81–6.

Costa M, Crowcroft J, Castro M, Rowstron A, Zhou L, Zhang L, Barham P. Vigilante: end-to-end containment of internet worms. SIGOPS Oper. Syst. Rev. 2005;39(5):133–47.

CVE. Common vulnerabilities and exposures-2008-2364. 〈http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2364〉, 2008.

Debian. Dsa-1571-1 Openssl: predictable random number generator. 〈http://www.debian.org/security/2008/dsa-1571〉, 2008.

Distler R, Kapitza R, Reiser HP. Efficient state transfer for hypervisor-based proactive recovery. In WRAITS '08: Proceedings of the 2nd workshop on recent advances on intrusiton-tolerant systems. ACM, New York, NY, USA, 2008. pp. 1–6.

Enisa. Cloud computing risk assessment. 〈http://www.enisa.europa.eu/act/rm/files/deliverables〉, 2009.

Enomaly. Enomalism. 〈http://www.enomaly.com〉, 2009.

Foster T, Zhao Y, Raicu I, Lu S. Cloud computing resource management through a grid middleware: A case study with diet and eucalyptus. Cloud Computing, IEEE International Conference on, 2009. pp. 151–4.

Gu L, Cheung S-C. Constructing and testing privacy-aware services in a cloud computing environment: challenges and opportunities. In Internetware '09: Proceedings of the first Asia-Pacific symposium on internetware. ACM New York, NY, USA, 2009. pp. 1–10.

Haeberlen A. A case for the accountable cloud. In LADIS '09: 3rd ACM SIGOPS International workshop on large scale distributed systems and middleware, 2009.

Hay B, Nance K. Forensics examination of volatile system data using virtual introspection. SIGOPS Oper. Syst. Rev. 2008;42(3):74–82.

Hohmuth M, Peter M, Härtig H, Shapiro JS. Reducing tcb size by using untrusted components: small kernels versus virtual-machine monitors. In EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop, ACM, New York, NY, USA, 2004. p. 22.

Honeynet Project. Sebek. 〈https://projects.honeynet.org/sebek/〉, 2003.

Huang Y, Arsenault D, Sood A. Closing cluster attack windows through server redundancy and rotations. In CCGRID, 2006. p. 21.

Huang Y, Geng X, Whinston AB. Defeating DDoS attacks by fixing the incentive chain. ACM Trans. Internet Technol. 2007;7(1):5.

Jiang X, Wang X, Xu D. Stealthy malware detection through vmm-based "out-of-the-box" semantic view reconstruction. In CCS '07: Proceedings of the 14th ACM conference on computer and communications security. ACM, New York, NY, USA, 2007. pp. 128–38.

Lombardi F, Di Pietro R. Transparent security for cloud. In SAC '10: Proceedings of the 2010 ACM symposium on applied computing (poster paper, to appear), ACM, New York, NY, USA, 2010.

Nurmi D, Wolski R, Grzegorczyk C, Obertelli G, Soman S, Youseff L, Zagorodnov D. The Eucalyptus open-source cloud-computing system. In CCGRID '09: Proceedings of the 2009 9th IEEE/ACM international symposium on cluster computing and the grid, IEEE Computer Society, Washington, DC, USA, 2009. pp. 124–31.

Openecp. Openecp. 〈http://www.openecp.org〉, 2010.

Payne BD, Carbone M, Sharif M, Lee W. Lares: An architecture for secure active monitoring using virtualization. In SP '08: Proceedings of the 2008 IEEE symposium on security and privacy (sp 2008), IEEE Computer Society, Washington, DC, USA, 2008. pp. 233–47.

Pearson S. Taking account of privacy when designing cloud computing services. In CLOUD '09: Proceedings of the 2009 ICSE workshop on software engineering challenges of cloud computing, IEEE Computer Society, Washington, DC, USA, 2009. pp. 44–52.

Perez R, van Doorn L, Sailer R. Virtualization and hardware-based security. IEEE Security and Privacy 2008;6(5):24–31.

Peter M, Schild H, Lackorzynski A, Warg A. Virtual machines jailed: virtualization in systems with small trusted computing bases. In VDTS '09: Proceedings of the 1st EuroSys Workshop on virtualization technology for dependable systems, ACM, New York, NY, USA, 2009. p. 18–23.

Pollitt M, Nance K, Hay B, Dodge RC, Craiger P, Burke P, Marberry C, Brubaker B. Virtualization and digital forensics: a research and education agenda. J. Digit. Forensic Pract. 2008;2(2):62–73.

Qumranet. Linux kernel virtual machine. 〈http://kvm.qumranet.com〉.

Ranadive A, Gavrilovska A, Schwan K. Ibmon: monitoring vmm-bypass capable infiniband devices using memory introspection. In HPCVirt '09: Proceedings of the 3rd ACM workshop on system-level virtualization for high performance computing, ACM, New York, NY, USA, 2009. p. 25–32.

RedHat. Libvirt. 〈http://libvirt.org〉, 2007.

Rellermeyer JS, Duller M, Alonso G. Engineering the cloud from software modules. In CLOUD '09: Proceedings of the 2009 ICSE workshop on software engineering challenges of cloud computing, IEEE Computer Society, Washington, DC, USA, 2009. p. 32–7.

Rhee J, Riley R, Xu D, Jiang X. Defeating dynamic data kernel rootkit attacks via vmm-based guest-transparent monitoring. Availability, Reliability and Security, 2009. ARES '09. International Conference on, 2009.

Riley R, Jiang X, Xu D. Guest-transparent prevention of kernel rootkits with vmm-based memory shadowing. In RAID '08: Proceedings of the 11th international symposium on recent advances in intrusion detection, Springer-Verlag, Berlin, Heidelberg, 2008. p. 1–20.

Ristenpart T, Tromert E, Shacham H, Savage S. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In CCS '09: Proceedings of the 14th ACM conference on computer and communications security, ACM, New York, NY, USA, 2009. p. 103–15.

Salza S, DiCarlo Y, Lombardi F, Puccinelli R. Leveraging the grid for the autonomic management of complex infrastructures. In: GCA grid computing and applications conference proceedings, 2006. p. 32–7.

Secunia. Secunia advisory. 〈http://secunia.com/advisories/36389〉, 2009.

Seshadri A, Luk M, Qu N, Perrig A. Secvisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity oses. In SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on operating systems principles, ACM, New York, NY, USA, 2007. p. 335–50.

Sam Johnston. Cve-2008-4990 enomaly ecp/enomalism: Insecure temporary file creation vulnerabilities. 〈http://www.securityfocus.com/archive/1/archive/1/500573/100/0/threaded〉, 2009.

Kim GH, Spafford EH. The design and implementation of tripwire: a file system integrity checker. In CCS '94: Proceedings of the 2nd ACM conference on computer and communications security. ACM, New York, NY, USA, 1994. pp. 18–29.

Lenk A, Klems M, Nimis J, Tai S, Sandholm T. What's inside the Cloud? An architectural map of the cloud landscape. In it CLOUD '09: Proceedings of the 2009 ICSE workshop on software engineering challenges of cloud computing, IEEE Computer Society, Washington, DC, USA, 2009. pp. 23–31.

Lombardi F, Di Pietro R. Kvmsec: a security extension for linux kernel virtual machines. In SAC '09: Proceedings of the 2009 ACM symposium on applied computing, ACM, New York, NY, USA, 2009. pp. 2029–34.

Lombardi F, Di Pietro R. A security management architecture for the protection of kernel virtual machines. In TSP '10: Proceedings of the Third IEEE international symposium on trust, security and privacy for emerging applications (to appear), IEEE Computer Society, Washington, DC, USA, 2010.

Siebenlist F. Challenges and opportunities for virtualized security in the clouds. In SACMAT '09: Proceedings of the 14th ACM symposium on access control models and technologies, ACM, New York, NY, USA, 2009. p. 1–2.

Smith B, Grehan R, Yager T. Byte-unixbench: A Unix benchmark suite. 〈http://code.google.com/p/byte-unixbench/〉.

Smith M, Friese T, Engel M, Freisleben B. Countering security threats in service-oriented on-demand grid computing using sandboxing and trusted computing techniques. J. Parallel Distrib. Comput. 2006;66(9):1189–204.

Sousa P, Bessani AN, Correia M, Neves NF, Verissimo P. Resilient intrusion tolerance through proactive and reactive recovery. Pacific Rim International Symposium on Dependable Computing, IEEE 2007;0:373–80.

Vaquero LM, Rodero-Merino L, Caceres J, Lindner M. A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 2009;39(1):50–5.